

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

NGUYỄN NHƯ' THẾ

**THIẾT KẾ THIẾT BỊ ĐO ĐỘ DÀI TỪ XA ỨNG DỤNG
CẢM BIẾN QUÁN TÍNH**

LUẬN VĂN THẠC SỸ KỸ THUẬT

Đà Nẵng - Năm 2018

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

NGUYỄN NHƯ' THỂ

THIẾT KẾ THIẾT BỊ ĐO ĐỘ DÀI TỪ XA ỨNG DỤNG
CẢM BIẾN QUÁN TÍNH

Chuyên ngành : KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

Mã số : 8520216

LUẬN VĂN THẠC SỸ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. Nguyễn Anh Duy

Đà Nẵng - Năm 2018

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả luận văn

Nguyễn Như Thế

MỤC LỤC

MỞ ĐẦU	9
1. Tính cấp thiết của đề tài	9
2. Tổng quan về hệ thống	10
3. Mục tiêu nghiên cứu	11
4. Đối tượng nghiên cứu và phạm vi nghiên cứu	11
4.1. Đối tượng nghiên cứu	11
4.2. Phạm vi nghiên cứu	11
5. Phương pháp nghiên cứu	11
5.1. Nghiên cứu lý thuyết	11
5.2. Nghiên cứu thực nghiệm	11
6. Bố cục luận văn	11
CHƯƠNG 1	12
GIỚI THIỆU CẢM BIẾN QUÁN TÍNH, LIDAR VÀ VI ĐIỀU KHIỂN ARDUINO	12
1.1. Giới thiệu cảm biến quán tính GY-85	12
1.2. Giới thiệu cảm biến khoảng cách LIDAR Lite V3	14
1.2.1. Tổng quan	14
1.2.2. Cảm biến khoảng cách LIDAR Lite V3	14
CHƯƠNG 2	15
THUẬT TOÁN ĐỊNH VỊ QUÁN TÍNH CƠ BẢN	15
2.1. Hệ trục tọa độ và một số khái niệm cơ bản	15
2.2. Quaternion	16
2.3. Quaternion trung bình	16
2.4. Biểu diễn góc sử dụng quaternion	17
2.5. Các phương trình vị trí và góc cơ bản	19
2.6. Phương trình góc quay cơ bản (góc euler)	20
2.7. Xác định góc quay từ giá trị đo	20
2.8. Các phương trình cảm biến	21
2.9. Ước lượng thay đổi và vị trí	21
2.9.1. Tính \mathbf{q}	22
2.9.2. Phương trình \mathbf{q}_e	22
2.9.3. Vận tốc và vị trí	23
2.10. AHRS	23

2.10.1 Bộ lọc cho AHRS.....	23
2.11. Xác định góc quay từ giá trị đo	26
2.11.1 Thuật toán TRIAD	26
2.11.2. Xác định góc quay từ cảm biến.....	26
2.11.3. Thuật toán tổng quát	26
2.11.4. Khởi tạo góc quay ban đầu sử dụng cảm biến từ trường và cảm biến gia tốc	28
CHƯƠNG 3.....	29
BỘ LỌC KALMAN	29
3.1 Giới thiệu chung về bộ lọc Kalman	29
3.2. Mô hình toán học	30
3.2.1. Hệ thống và mô hình quan sát	30
3.2.2. Giả thiết	31
3.2.3. Nguồn gốc.....	31
3.2.4. Điều kiện không chệch	32
3.2.5. Hiệp phương sai sai số.....	33
3.2.6. Độ lùi Kalman	34
3.2.7. Tóm tắt các phương trình của bộ lọc Kalman	35
CHƯƠNG 4.....	37
THUẬT TOÁN XÁC ĐỊNH THÔNG SỐ CỦA CẢM BIẾN VÀ THUẬT TOÁN ĐỊNH VỊ QUÁN TÍNH CHO BỘ LỌC KALMAN.....	37
4.1. Thuật toán xác định các thông số của cảm biến khoảng cách.....	37
4.2. Thuật toán định vị quán tính (ina) sử dụng bộ lọc kalman	38
4.3. Ứng dụng bộ lọc Kalman cho thuật toán định vị quán tính	40
CHƯƠNG 5.....	42
THIẾT KẾ PHẦN CỨNG, THỰC NGHIỆM ĐO KHOẢNG CÁCH VÀ ĐÁNH GIÁ KẾT QUẢ ĐO.....	42
5.1. Sơ đồ chi tiết.....	42
5.2. Đọc dữ liệu từ LIDAR.....	42
5.3. Đọc dữ liệu từ GY-85.....	43
5.4. Lập trình cho vi điều khiển	45
5.5. Viết chương trình cho phần mềm Matlab	45
5.6. Cài đặt điểm đo	45
5.7. Thực hiện thu thập dữ liệu	46
5.8. Thống kê kết quả đo	47
5.9. Đánh giá sai lệch	48
5.10. Nguyên nhân sai số	49

KẾT LUẬN VÀ KIẾN NGHỊ.....50

TÀI LIỆU THAM KHẢO.....51

PHỤ LỤC 152

PHỤ LỤC 2.....57

ĐO ĐỘ DÀI TỪ XA ỨNG DỤNG CẢM BIẾN QUÁN TÍNH VÀ CẢM BIẾN ĐO XA

Học viên: Nguyễn Như Thế

Chuyên ngành: Kỹ thuật điều khiển và tự động hóa

Mã số: 60.52.02.16

Khóa: 33

Trường Đại học Bách khoa – ĐHQGHN

Tóm tắt - Cảm biến quán tính (Inertial Measurement Unit – IMU) và cảm biến khoảng cách hiện đang được sử dụng rất rộng rãi trong nhiều lĩnh vực của kỹ thuật và đời sống. Trong phần trình bày tác giả thiết kế một hệ thống đo xa sử dụng một cảm biến khoảng cách dùng laser kết hợp với một cảm biến quán tính (IMU). Bằng cách kết hợp phân tích và tính toán quỹ đạo chuyển động của hệ thống (được ước lượng từ dữ liệu của cảm biến quán tính) và khoảng cách từ hệ thống đến các điểm cần đo (đo được từ cảm biến khoảng cách), ta có thể tính được tọa độ không gian của các điểm cần đo. Từ các dữ liệu đó có thể tính toán các thông số hình học khác như khoảng cách, góc, diện tích... trong không gian trong phạm vi cho phép. Kết quả thực nghiệm cho thấy khả năng ứng dụng thực tế của hệ thống được đề xuất.

Từ khóa – Đo khoảng cách từ xa; cảm biến khoảng cách; cảm biến quán tính; khối định vị quán tính – IMU; bộ lọc Kalman.

Abstract –The Inertial Measurement Unit (IMU) and distance measuring sensor is now widely used in many areas of technology and life. In the presentation, the author designed a remote measurement system using a laser distance sensor combined with an inertial sensor (IMU). By combining the analysis and calculation of the motion trajectory of the system (estimated from inertial sensor data) and the distance from the system to the points to be measured (measured from distance sensors), We can calculate the spatial coordinates of the points to be measured. From the data it can calculate other geometric parameters such as distance, angle, area... in the space within the allowed range. Experimental results show that the practical applicability of the system is proposed.

Key word – Remote measuring; distance measuring sensor; inertial sensor; inertial measuring unit – IMU; Kalman filter.

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

LCD	Lyquify Crystal Display
BCS	Body Coordinate System
WCS	World Coordinate System
IMU	Inertial Measurement Unit
LIDAR	Laser Imaging Detection And Ranging
IDE	Integrated Development Environment
CPU	Center Processsing Unit
INA	Inertial Navigation Algorithm
ZVI	Zero Velocity Interval

DANH MỤC CÁC BẢNG

Số hiệu	Tên bảng	Trang
5.1	Thống kê kết quả đo thực nghiệm	48
5.2	Biểu đồ thống kê sai lệch độ dài AB	49
5.3	Biểu đồ thống kê sai lệch độ dài BC	49
5.4	Biểu đồ thống kê sai lệch độ dài AC	50
5.5	Bảng thống kê sai lệch nhỏ nhất và lớn nhất từng đoạn AB, BC, AC	50

DANH MỤC CÁC HÌNH

Số hiệu	Tên hình	Trang
1	Tổng quan hệ thống đề xuất	10
1.1	Hình ảnh IMU	12
1.2	Các cấu trúc của khối IMU vi cơ.	13
1.3	Cảm biến quán tính GY-85	13
1.4	Cảm biến khoảng cách LIDAR lite	14
2.1	Hệ trục tọa độ dẫn hướng và khối tâm	15
3.1	Mô hình đo lường ước lượng của bộ lọc Kalman	30
3.2	Tín hiệu thu trước và sau khi lọc qua Kalman	31
3.3	Mô hình không gian trạng thái	31
3.4	Tóm tắt quá trình khởi tạo của Kalman	37
4.1	Phương pháp xác định các thông số của cảm biến khoảng cách	38
4.2	Vec tơ vị trí \vec{r} khi không sử dụng bộ lọc Kalman	41
4.3	Vec tơ vị trí \vec{r} khi có sử dụng bộ lọc Kalman	42
5.1	Sơ đồ thiết bị đo khoảng cách từ xa	43
5.2	Lưu đồ thuật toán đọc dữ liệu từ LIDAR	44
5.3	Lưu đồ thuật toán chương trình đọc dữ từ GY-85	45
5.4	Các điểm cần đo trên bức tường	46
5.5	Thiết bị đo độ dài từ xa	47
5.6	Dữ liệu truyền qua giao diện serial monitor	48

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Đo khoảng cách là vấn đề cực kỳ quan trọng trong nhiều lĩnh vực như cơ khí, quân sự, y tế, nghiên cứu khoa học, đặc biệt là lĩnh vực đo khoảng cách trong xây dựng, trắc địa... đem lại lợi ích to lớn với đời sống của chúng ta.

Trước đây, việc đo đạc kết quả đưa ra thường không chuẩn xác và mất nhiều thời gian, kinh phí nhân công cao, năng suất làm việc thấp vì sử dụng thiết bị thô sơ, việc đo khoảng cách sẽ khó khăn hơn đối với những vị trí khó chạm tay đến như hố sâu, độ sâu của hồ, giếng, biển, các tòa nhà cao tầng, trong nhà máy hóa chất, khoảng cách trong lĩnh vực quân sự

Ngày nay với sự phát triển nhanh chóng và không ngừng của khoa học kỹ thuật, đặc biệt là việc ứng dụng các công nghệ mới, thiết bị vi điều khiển kết hợp với cảm biến ngày càng phổ biến trong cuộc sống, với mọi lĩnh vực, phục vụ cho cuộc sống con người ngày càng tốt hơn. Một trong những ứng dụng trong lĩnh vực xây dựng, quân sự được nhiều người quan tâm đến đó là tạo ra một thiết bị đo khoảng cách từ xa, giúp cho việc đo khoảng cách trở nên nhanh chóng và chính xác hơn. Để đo khoảng cách từ vị trí đang đứng đến điểm cần đo thì chỉ cần đứng yên và trỏ tia laser đến một điểm cần đo, kết quả sẽ hiển thị lên màn hình LCD một cách nhanh chóng, tất cả các tính toán của phép đo đều được thao tác trực tiếp nhanh chóng thông qua các phím, màn hình của thước đo, hơn nữa bạn cũng có thể lưu trữ nhiều kết quả đo để phục vụ cho công việc sau này.

Các loại cảm biến sử dụng cho việc đo khoảng cách phổ biến đó là cảm biến siêu âm, cảm biến ánh sáng, cảm biến laser. Tuy nhiên, các loại cảm biến này chỉ đo được khoảng cách trực tiếp từ vị trí đo đến điểm cần đo mà không thể đo được khoảng cách giữa các điểm tùy ý trong không gian. Do vậy, việc ứng dụng cảm biến để nghiên cứu phát triển thiết bị đo khoảng cách giữa hai vị trí tùy ý trong không gian là cần thiết.

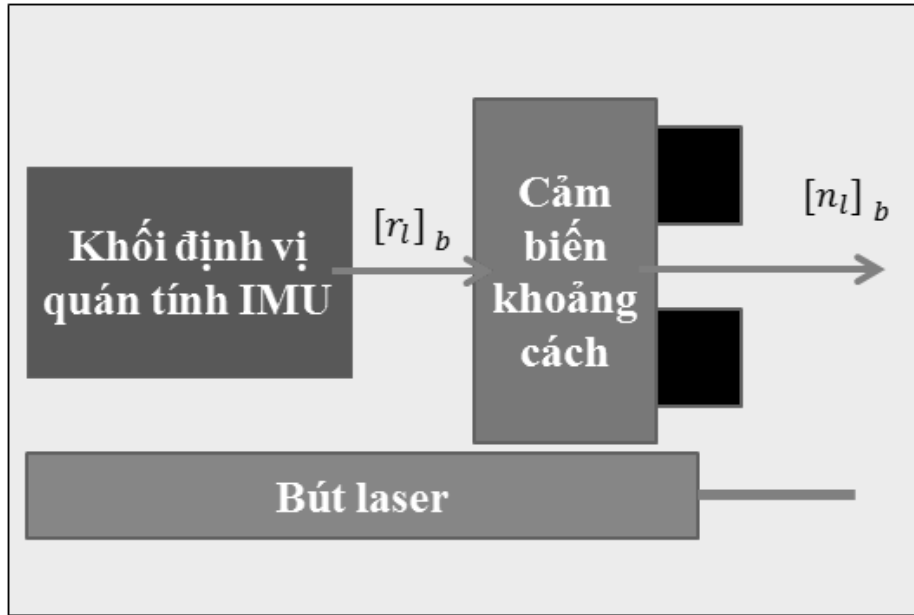
Hiện nay, cảm biến quán tính phát triển rất nhanh và đa dạng, được ứng dụng rộng rãi vào hệ thống định vị dẫn đường, hầu hết các điện thoại thông minh, các ứng dụng định vị, chuyển đổi màn hình ngang sang màn hình dọc, theo dõi sức khỏe, các trò chơi... Việc ứng dụng cảm biến quán tính kết hợp với cảm biến laser để nghiên cứu và thực nghiệm đo khoảng giữa hai vị trí trong không gian là ý tưởng hay.

Cũng chính từ những nguyên nhân trên, tôi quyết định chọn đề tài “THIẾT KẾ THIẾT BỊ ĐO ĐỘ DÀI TỪ XA ỨNG DỤNG CẢM BIẾN QUÁN TÍNH”

” nhằm nghiên cứu kỹ hơn về thuật toán định vị quán tính, bộ lọc Kalman, và ứng dụng của cảm biến quán tính, cảm biến laser để thực nghiệm việc đo khoảng cách từ xa. Trong đó, cảm biến laser sử dụng để đo khoảng cách từ hệ thống đến các điểm cần đo trong khi cảm biến quán tính được sử dụng để ước lượng sự chuyển động của hệ thống trong quá trình đo. Từ việc kết hợp các khoảng cách đo và quỹ đạo chuyển động của hệ thống, ta có thể ước lượng được khoảng cách giữa hai điểm cần đo.

2. Tổng quan về hệ thống

Hệ thống đo độ dài từ xa được đề xuất (Hình 1) bao gồm một cảm biến khoảng cách bằng laser (Laser-lite sensor, công ty PulsedLight Inc., Bend) và một cảm biến quán tính. Cảm biến khoảng cách này hoạt động theo nguyên tắc tính thời gian truyền đến đối tượng và phản xạ lại của tia laser (time-of-flight) với phạm vi đo đến 40m và tần số lấy mẫu là 33,33Hz. Cảm biến quán tính bao gồm cảm biến gia tốc và cảm biến vận tốc góc theo 3 trục với tần số lấy mẫu là 100 Hz. Một bút laser được gắn vào hệ thống để hiển thị điểm mà hệ thống đang chỉ vào. Do kích thước nhỏ, gọn nên hệ thống có thể được sử dụng như là một thiết bị cầm tay. Bằng cách chỉ hệ thống vào một điểm và di chuyển để hệ thống chỉ vào điểm khác, chúng ta có thể ước lượng được mối quan hệ (độ dài, độ cao...) giữa 2 điểm đó cũng như có thể mở rộng đối với những thông tin khác giữa nhiều điểm.



Hình 1: Tổng quan hệ thống đề xuất

Hệ thống sử dụng một hệ trục tọa độ biến đổi và một hệ trục tọa độ cố định. Hệ trục tọa độ biến đổi hay còn gọi là hệ tọa độ người dùng được sử dụng gắn liền với hệ thống (BCS - Body Coordinate System) và được chọn trùng với hệ trục tọa độ của cảm biến quán tính. Hệ trục tọa độ cố định hay còn gọi là hệ trục tọa độ toàn cầu (WCS – World Coordinate System). Sự chuyển động của hệ thống được đề xuất sẽ được biểu diễn trong WCS. Trục z của WCS hướng lên trên theo phương thẳng đứng (trùng với phương của gia tốc trọng trường). Góc và phương x của WCS không ảnh hưởng đến quá trình đo các thông số (độ dài, độ cao, góc và diện tích) nên được chọn tùy ý. Để phân biệt tọa độ đang xét trong hệ nào trong trường hợp dễ xảy ra hiểu nhầm, bài báo này sử dụng chỉ số dưới. Ví dụ: $[p]_b$ (hay $[p]_w$) được sử dụng để thể hiện vector $p \in R^3$ được biểu diễn trong hệ tọa độ BCS (hay trong hệ tọa độ WCS).

Việc xác định mối quan hệ về vị trí và hướng của cảm biến khoảng cách và cảm biến quán tính là rất cần thiết trong hệ thống đề xuất. Trong Hình 1, $[r_l]_b \in R^3$ thể hiện vị trí của cảm biến khoảng cách đối với cảm biến quán tính và $[n_l]_b$ thể hiện hướng của cảm biến khoảng cách đối với cảm biến quán tính. Những thông số này cần được ước lượng thông qua thuật toán xác định thông số của cảm biến khoảng cách ở mục tiếp theo.

3. Mục tiêu nghiên cứu

Ứng dụng được thuật toán định vị quán tính sử dụng bộ lọc Kalman trong ước lượng chuyển động cho cảm biến quán tính.

Xây dựng được hệ thống đo khoảng cách từ xa một cách linh hoạt với độ chính xác cho phép phục vụ cuộc sống kỹ thuật.

4. Đối tượng nghiên cứu và phạm vi nghiên cứu

4.1. Đối tượng nghiên cứu

- Cảm biến quán tính
- Đo khoảng cách giữa hai vị trí trong không gian
- Thuật toán định vị quán tính
- Lý thuyết bộ lọc Kalman
- Phần mềm Matlab xử lý và tính toán dữ liệu từ vi điều khiển

4.2. Phạm vi nghiên cứu

- Đề tài này sử dụng cảm biến quán tính GY-85, sử dụng cảm biến đo xa LIDAR Lite. Phân tích dữ liệu offline trên máy tính với board vi điều khiển arduino UNO, chỉ dừng lại ở việc ước lượng được khoảng cách mà không phân tích, đánh giá sai số.

5. Phương pháp nghiên cứu

5.1. Nghiên cứu lý thuyết

- Tìm hiểu cơ sở lý thuyết của việc đo khoảng cách từ xa qua thông qua các bài báo, internet, sách...

- Tham khảo các hệ thống, thiết bị liên quan đã có trong thực tế

- Xây dựng thuật toán định vị quán tính sử dụng bộ lọc Kalman để ước lượng chuyển động cho hệ thống. Kết hợp quỹ đạo chuyển động của hệ thống với khoảng cách đo từ cảm biến khoảng cách để tính vị trí 3D của các điểm từ xa. Từ vị trí 3D của các điểm này, ta tính khoảng cách tương đối giữa chúng

5.2. Nghiên cứu thực nghiệm

- Tiến hành chế tạo sản phẩm và thí nghiệm đo khoảng cách để chứng minh, đánh giá độ chính xác

6. Bố cục luận văn

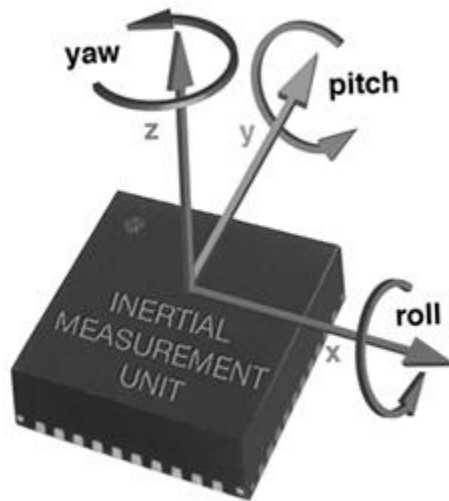
- Chương 1: Giới thiệu cảm biến quán tính, lidar và vi điều khiển arduino
 - Chương 2: Thuật toán định vị quán tính cơ bản
 - Chương 3: Bộ lọc Kalman
 - Chương 4: Thuật toán xác định các thông số của cảm biến khoảng cách và thuật toán định vị quán tính sử dụng bộ lọc Kalman
 - Chương 5: Thiết kế phần cứng, thực nghiệm đo khoảng cách và đánh giá kết quả đo
- KẾT LUẬN VÀ KIẾN NGHỊ

CHƯƠNG 1

GIỚI THIỆU CẢM BIẾN QUÁN TÍNH, LIDAR VÀ VI ĐIỀU KHIỂN ARDUINO

1.1. Giới thiệu cảm biến quán tính GY-85

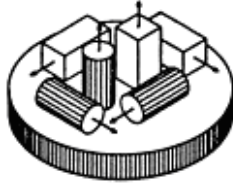
Khối IMU thời kì đầu sử dụng những cảm biến quán tính hoạt động theo nguyên tắc cơ khí. Những cảm biến cơ khí này thường có kích thước lớn, hoạt động kém hiệu quả, giá thành cao và tiêu thụ nhiều năng lượng. Ngày nay, cùng với sự tiến bộ của khoa học công nghệ, đặc biệt là công nghệ vật liệu mới và công nghệ vi chế tạo đã tạo ra các cảm biến vi cơ có kích thước rất nhỏ (cỡ centimet (Hình 2.1)), hoạt động hiệu quả, tiêu thụ ít năng lượng và đặc biệt là giá thành hạ, điều này mở ra một khả năng rộng lớn cho việc ứng dụng các cảm biến vi cơ trong nhiều lĩnh vực đời sống.



Hình 1.1: Hình ảnh IMU

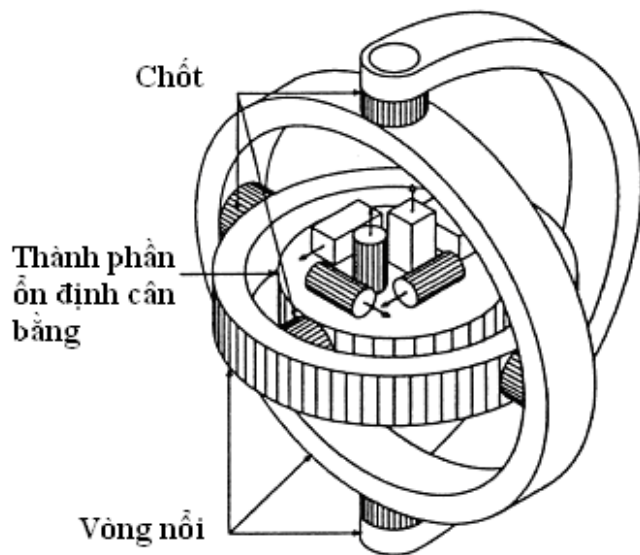
Một khối vi cơ IMU được cấu tạo từ các cảm biến vi cơ, thường là 3 cảm biến gia tốc và 3 cảm biến vận tốc góc, hoặc cũng có thể là 1 cảm biến gia tốc 3 chiều kết hợp với 3 cảm biến vận tốc góc. Các cảm biến vi cơ kết cấu hỗ trợ với nhau theo cấu trúc gắn liền (Hình 1.2a) hoặc theo cấu trúc nổi (Hình 1.2b), từ đó có thể xác định được 3 thành phần chuyển động quay và tịnh tiến của vật thể.

Tổ hợp cảm biến gia tốc
và cảm biến vận tốc góc



Mô hình gắn chặt
với vật thể cần đo

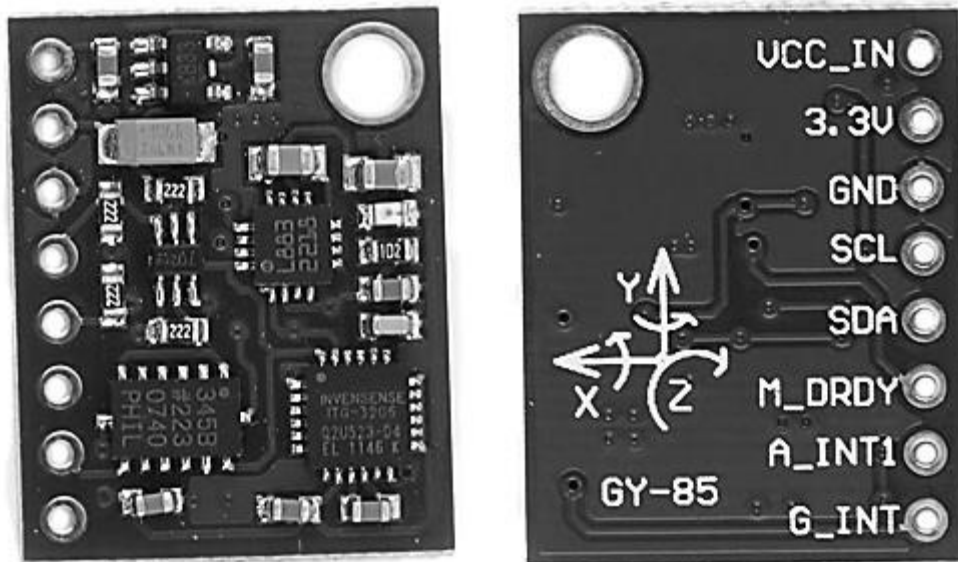
(a) Gắn liền



(b) Nổi

Hình 1.2 Các cấu trúc của khối IMU vi cơ.

Điểm khác nhau cơ bản của hai kiểu cấu trúc này đó là: Với kiểu nổi (Gimbal) thì các cảm biến bị thay đổi hướng theo đối tượng chuyển động; còn trong kiểu gắn chặt (Strapdown) thì các cảm biến được gắn chặt với vật chuyển động, do đó sẽ không thay đổi trạng thái chuyển động theo vật đó. Trên thực tế khối IMU có cấu trúc kiểu gắn chặt được sử dụng rộng rãi hơn bởi cấu trúc này đơn giản và có giá thành chế tạo thấp với độ chính xác có thể chấp nhận được.



Hình 1.3: Cảm biến quán tính GY-85

Khi kết hợp các cảm biến vi cơ thành một cấu trúc tổng thể thì thường tạo ra sai số. Sai số mắc phải trong việc sử dụng các cảm biến vi cơ này có ở 2 cấp độ, cấp độ cảm biến và cấp độ nhóm cảm biến. Ở cấp độ cảm biến là sai số của từng cảm biến cấu tạo nên khối IMU, còn ở cấp độ nhóm cảm biến là sai số tổ hợp của nhóm cảm biến với nhau.

1.2. Giới thiệu cảm biến khoảng cách LIDAR Lite V3

1.2.1. Tổng quan

Lidar là một công nghệ khảo sát tiên tiến để đo khoảng cách tới mục tiêu bằng cách chiếu mục tiêu đó bằng một tia laser và đo các xung phản xạ bằng một cảm biến. Sự khác nhau về thời gian và bước sóng laser sau đó có thể được sử dụng để tạo các mô hình kỹ thuật số thể hiện mục tiêu trong không gian 3D. LIDAR thường được kết hợp với các loại máy ảnh để thu thập đồng thời hình ảnh của đối tượng và xây dựng lên hình ảnh 3D “như thật” của đối tượng. Lidar được sử dụng rộng rãi để tạo ra các sản phẩm đồ họa có độ phân giải cao, với nhiều ứng dụng cao cấp trong các ngành trắc địa, geomatics, khảo cổ, địa lý, địa chất, địa mạo, địa chấn, lâm nghiệp, vật lý, bản đồ... Công nghệ này cũng được sử dụng để kiểm soát và điều hướng cho một số phương tiện giao thông tự hành. Lidar đôi khi được gọi là quét laser và quét 3D, và gồm 3 loại: lidar mặt đất, hàng không và di động.

1.2.2. Cảm biến khoảng cách LIDAR Lite V3

PulsedLight nhằm mục tiêu sự cần thiết của các cảm biến đo khoảng cách quang học hiệu suất cao, rất nhỏ gọn cho các ứng dụng như robot và UAV, nơi có một cảm biến quang học rất nhỏ, năng lượng thấp, hiệu năng cao và giảm chi phí



Hình 1.4: Cảm biến khoảng cách LIDAR lite

Việc sử dụng kỹ thuật kết hợp ký hiệu (được gọi là tương quan tín hiệu) ước tính độ trễ thời gian bằng cách trượt điện tử một tham chiếu truyền tải được lưu giữ qua tín hiệu thu lại để tìm ra kết quả tốt nhất.

Hoạt động của laser trong các vụ nổ ngắn cho phép khuếch đại 100: 1 với công suất ra cực đại trên các hệ thống đo bằng một chùm liên tục

CHƯƠNG 2

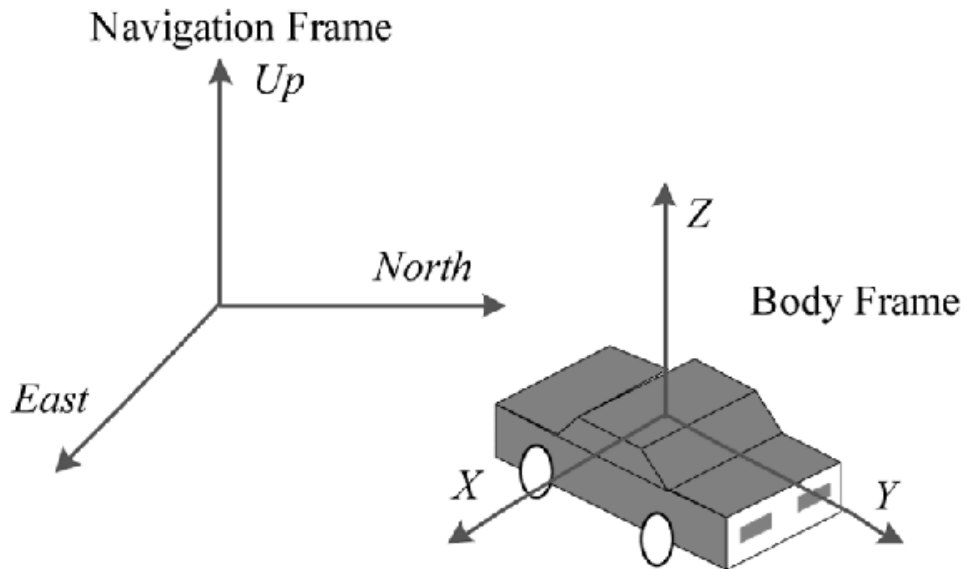
THUẬT TOÁN ĐỊNH VỊ QUẢN TÍNH CƠ BẢN

2.1. Hệ trục tọa độ và một số khái niệm cơ bản

Vì chúng ta chỉ quan tâm đến một hệ thống dẫn đường quán tính rất ngắn, chỉ cần xem xét hai tọa độ: hệ trục dẫn hướng và hệ trục tọa độ người dùng.

Hệ trục dẫn hướng: là một hệ trục địa lý cục bộ mà có trục hướng bắc (trục x), hướng tây (trục y), và trục đứng cục bộ (trục z)

Hệ trục khối tâm: một hệ trục mà các trục trùng với 3 trục trong đơn vị dẫn đường quán tính. Đối với một vector xác định p chúng ta thường sử dụng chỉ số b (body) và n (navigation) để nhấn mạnh rằng một vec tơ được biểu diễn trong một hệ trục tọa độ cụ thể.



Hình 2.1: Hệ trục tọa độ dẫn hướng và khối tâm

- Ký hiệu sử dụng trong hệ thống quán tính
- $r_n \in R^{3 \times 1}$: vị trí trong hệ tọa độ dẫn hướng
- $v_n \in R^{3 \times 1}$: vận tốc trong hệ tọa độ dẫn hướng
- $\omega_b \in R^{3 \times 1}$: góc quay khối tâm
- $a_b \in R^{3 \times 1}$: gia tốc được sinh ra từ các lực khác và từ trường hấp dẫn
- $g \in R^{3 \times 1}$: gia tốc trọng lực trong hệ tọa độ dẫn hướng
- $y_g \in R^3$: đầu ra vận tốc góc
- $b_g \in R^3$: hướng vận tốc góc
- $v_g \in R^3$: nhiễu đo lường vận tốc góc
- $y_a \in R^3$: đầu ra gia tốc

- $b_a \in R^3$: hướng gia tốc
- $v_a \in R^3$: nhiễu đo lường gia tốc
- $q \in R^4$: quaternion đại diện góc quay từ trục tọa độ dẫn hướng đến trục tọa độ khối tâm
- $\hat{q} \in R^4$: tích phân quaternion
- $q_e \in R^4$: lỗi quaternion trong q, $q = \hat{q} \otimes q_e$
- $K(a)$ được định nghĩa bởi ($\in R^3$)

$$K(a) = \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

- Sản phẩm chéo

$$a \times b = K(a)b = -b \times a = -K(b)a$$

2.2. Quaternion

Chúng ta sử dụng quaternion cho biểu diễn phản ứng

Quaternion $q_n^b = [q_0 \ q_1 \ q_2 \ q_3]$ là vec tơ định nghĩa phản ứng

Đề cho p và q là những quaternion.

$$p = [p_0 \ \vec{p}], \quad q = [q_0 \ \vec{q}]$$

mà

$$\bar{p} = [p_1 \ p_2 \ p_3], \quad \bar{q} = [q_1 \ q_2 \ q_3]$$

- Sản phẩm chéo theo công thức (2.1) [6]

$$p \otimes q = p_0 q_0 - \bar{p} \cdot \bar{q} + p_0 \bar{q} + \bar{p} \times \bar{q} \quad (2.1)$$

Phương trình dưới dạng ma trận : $r = pq$ theo công thức (2.2) [6]

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (2.2)$$

Quaternion biểu diễn một khối quay của góc ϕ về vec tơ đơn vị \bar{u} được cho bởi công thức (2.3) [6]

$$q = \left[\cos\left(\frac{\phi}{2}\right) \quad \sin\left(\frac{\phi}{2}\right)\bar{u} \right] \quad (2.3)$$

Quaternion biểu diễn dưới dạng ma trận theo công thức (2.4) [6]

$$pq = \begin{bmatrix} p_0 & \bar{p}' \\ \bar{p} & p_0 I + [\bar{p} \times] \end{bmatrix} \begin{bmatrix} q_0 \\ \bar{q} \end{bmatrix} = \begin{bmatrix} q_0 & \bar{q}' \\ \bar{q} & q_0 I + [\bar{q} \times] \end{bmatrix} \begin{bmatrix} p_0 \\ \bar{p} \end{bmatrix} \quad (2.4)$$

2.3. Quaternion trung bình

- Vấn đề cơ bản : Tìm \bar{q} , quaternion trung bình của q_i
- Giả thiết

$$q = q_{e,i} \otimes q_i$$

với

$$R_i = E\{q_{e,i} q'_{e,i}\}$$

$$\bar{q}_{e,i} = q \otimes \bar{q}_i^* = \begin{bmatrix} q_{i,0} & \bar{q}'_i \\ -\bar{q}_i & \bar{q}_{i,0}I + [\bar{q}_i \times] \end{bmatrix} q$$

$$-\bar{q}_{e,i} = \Xi q$$

Trong đó Ξ được định nghĩa bằng

$$\Xi(q_i) = [-\bar{q}_i \quad q_{i,0}I + [\bar{q}_i \times]]$$

vấn đề tối ưu

$$\bar{q} = \min_q \sum_{i=1}^N q'_{e,i} R_i^{-1} q_{e,i} = \min_q \sum_{i=1}^N q' \Xi(q_i)' \Xi(q_i) q = \min_q q' M q$$

mà

$$M = \sum_{i=1}^N \Xi(q_i)' R_i^{-1} \Xi(q_i)$$

Giải pháp tối ưu tương ứng với một đặc trưng riêng với giá trị riêng tối đa của $-M$ sai số sai lệch của \bar{q}

$$\bar{R}^{-1} = \Xi(\bar{q}) \left(\sum_{i=1}^N \Xi(q_i)' R_i^{-1} \Xi(q_i) \right) \Xi(\bar{q})'$$

Công cụ ước lượng UISL : quaternion trung bình

Lệnh sử dụng trong Matlab: [qbar,Rbar] = quaternionaveraging2(q,R)

Nếu số lượng quaternion là N, thì kích thước của q và R là $4 \times N$ và $3 \times 3 \times N$

2.4. Biểu diễn góc sử dụng quaternion

P_b và p_n liên quan với nhau như biểu thức sau [6]

$$p_b = C_n^b(q) p_n = q^* p_n q$$

Với $q^* = [q_0 \quad -q_1 \quad -q_2 \quad -q_3]$

Quan hệ giữa q và ma trận quay tương ứng được cho bởi

$$C_n^b(q) = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}$$

DCM đến quaternion : đơn giản

-TrC được cho bởi

$$TrC = 6q_0^2 + 2(q_1^2 + q_2^2 + q_3^2) - 3 = 6q_0^2 + 2(1 - q_0^2) - 3 = 4q_0^2 - 1$$

Do đó q_0 được cho bởi

$$q_0 = 0.5 * \sqrt{TrC + 1}$$

- Tính toán $C' - C$ theo công thức (2.5) [6]

$$C' - C = 4q_0 K(\bar{q}) \Rightarrow K(\bar{q}) = \frac{1}{4q_0} (C' - C) \quad (2.5)$$

-Chuyển DCM sang quaternion : thuật toán Shepperd's

+ Tìm q_j^2 như sau:

$$q_0^2 = \frac{1}{4} (1 + TrC)$$

$$q_1^2 = \frac{1}{4} (C_{11} - C_{22} - C_{33} + 1)$$

$$q_2^2 = \frac{1}{4} (C_{22} - C_{11} - C_{33} + 1)$$

$$q_3^2 = \frac{1}{4} (C_{33} - C_{11} - C_{22} + 1)$$

- Chọn q_j lớn nhất

- Chúng ta có thể tính toán quaternion như sau

$$C_{12} + C_{21} = 4q_1q_2$$

$$C_{13} + C_{31} = 4q_1q_3$$

$$C_{23} + C_{32} = 4q_2q_3$$

$$C_{12} - C_{21} = 4q_0q_3$$

$$C_{13} - C_{31} = -4q_0q_2$$

$$C_{23} - C_{32} = 4q_0q_1$$

* Sử dụng công cụ ước lượng UISL : Tính toán q từ C

Lệnh sử dụng trong Matlab: $q = dcm2quaternion(C)$

- Biểu diễn góc Euler : $C_n^b = C_{x'}(\phi)C_{y'}(\theta)C_z(\psi)$ theo công thức (2.6) [6]

$$C_n^b(q) = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \quad (2.6)$$

Mà

$$C_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

$$C_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$C_z(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Các góc Euler với quaternion

$$\begin{aligned} q_0 &= \cos\frac{\psi}{2} \cos\frac{\theta}{2} \cos\frac{\phi}{2} + \sin\frac{\psi}{2} \sin\frac{\theta}{2} \sin\frac{\phi}{2} \\ q_1 &= \cos\frac{\psi}{2} \cos\frac{\theta}{2} \sin\frac{\phi}{2} - \sin\frac{\psi}{2} \sin\frac{\theta}{2} \cos\frac{\phi}{2} \\ q_2 &= \cos\frac{\psi}{2} \sin\frac{\theta}{2} \cos\frac{\phi}{2} + \sin\frac{\psi}{2} \cos\frac{\theta}{2} \sin\frac{\phi}{2} \\ q_3 &= \sin\frac{\psi}{2} \cos\frac{\theta}{2} \cos\frac{\phi}{2} - \cos\frac{\psi}{2} \sin\frac{\theta}{2} \sin\frac{\phi}{2} \end{aligned}$$

Quaternion với các góc Euler theo công thức (2.7)

$$\begin{aligned} \psi &= \text{atan2}(2q_1q_2 + 2q_0q_3, 2q_0^2 + 2q_1^2 - 1) \\ \theta &= -\sin^{-1}(2q_1q_3 - 2q_0q_2) \\ \phi &= \text{atan2}(2q_2q_3 + 2q_0q_1, 2q_0^2 + 2q_3^2 - 1) \end{aligned} \quad (2.7)$$

Khi góc quay nhỏ $\Rightarrow q_0 \approx 1, q_1, q_2, q_3 \approx 0$ góc Euler được theo công thức sau:

$$\begin{aligned} C(q) &\approx \begin{bmatrix} 1 & 2q_3 & -2q_2 \\ -2q_3 & 1 & 2q_1 \\ 2q_2 & -2q_1 & 1 \end{bmatrix} \\ &= I - 2 \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \end{aligned} \quad (2.8)$$

2.5. Các phương trình vị trí và góc cơ bản

Các phương trình cho định vị quán tính được cho theo công thức (2.9) [6] như sau:

$$\begin{aligned} \dot{q} &= \frac{1}{2} \Omega(\omega_b) q \\ \dot{v}_n &= a_n = C'(q) a_b \\ \dot{r}_n &= v_n \end{aligned} \quad (2.9)$$

Mà

$$\begin{aligned} \Omega(\omega) &\triangleq \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \\ q_n^b(t) &= q_n^{b(t)} \end{aligned}$$

do đó phương trình góc quay theo công thức:

$$q_n^b(t+dt) = q_n^{b(t+dt)} = q_n^{b(t)} q_{b(t)}^{b(t+dt)} \quad (2.10)$$

Trong khoảng thời gian dt , $b(t)$ sẽ quay một góc bởi

$$\phi = \|\omega_b\|_2 dt$$

Do đó từ phương trình (2.3), ta có:

$$q_{b(t)}^{b(t+dt)} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) & \sin\left(\frac{\phi}{2}\right) \frac{\omega_b}{\|\omega_b\|_2} \end{bmatrix}$$

Nếu dt nhỏ, vậy $\frac{\phi}{2} = \frac{\|\omega_b\|_2 dt}{2}$. Do đó ta có thể tính gần đúng phương trình trên như sau (sử dụng gần đúng $\cos\theta \approx 1, \sin\theta \approx \theta$ khi θ nhỏ):

$$q_{b(t)}^{b(t+dt)} \approx \begin{bmatrix} 1 & \frac{\|\omega_b\|_2 dt}{2} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\omega_b dt}{2} \end{bmatrix} \quad (2.11)$$

Thế phương trình (2.11) vào phương trình (2.10), ta có

$$q_n^b(t+dt) = q_n^b(t) \begin{bmatrix} 1 & \frac{\omega_b}{2} \end{bmatrix} = q_n^b(t) + q_n^b(t) \begin{bmatrix} 0 & \frac{\omega_b}{2} \end{bmatrix}$$

Có thể viết phương trình như sau:

$$\frac{q_n^b(t+dt) - q_n^b(t)}{dt} = q_n^b(t) \begin{bmatrix} 0 & \frac{\omega_b}{2} \end{bmatrix}$$

Nếu dt gần bằng 0, ta có

$$\dot{q}_n^b(t) = q_n^b(t) \begin{bmatrix} 0 & \frac{\omega_b}{2} \end{bmatrix}$$

2.6. Phương trình góc quay cơ bản (góc euler)

Phương trình trạng thái cơ bản sử dụng góc Euler theo công thức (2.12) [6]:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.12)$$

2.7. Xác định góc quay từ giá trị đo

Đo gia tốc: $y_a \Rightarrow$ hiệu chỉnh gia tốc $y_{a,c}$

$$y_{a,c} \triangleq \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} (y_a - y_b)$$

Tìm k_x, k_y, k_z , và b

$$\begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} (y_a - y_b) = C(q) \tilde{g}$$

Thực tế sử dụng $\|C(q)\tilde{g}\| = g^2$, ta có

$$k_x^2(y_{a,x} - b_x)^2 + k_y^2(y_{a,y} - b_y)^2 + k_z^2(y_{a,z} - b_z)^2 = g^2$$

$$y_{a,x}^2 k_x^2 - 2y_{a,x} b_x k_x^2 + y_{a,y}^2 k_y^2 - 2y_{a,y} b_y k_y^2 + k_y^2 b_y^2 + y_z^2 k_z^2 - 2y_{a,z} b_z k_z^2 + k_z^2 b_z^2 - g^2 = 0$$

$$Ap = 0$$

Mà

$$A = \begin{bmatrix} y_{a,x}^2 & y_{a,y}^2 & y_{a,z}^2 & -2y_{a,x} & -2y_{a,y} & -2y_{a,z} & 1 \end{bmatrix}$$

$$p = \begin{bmatrix} k_x^2 \\ k_y^2 \\ k_z^2 \\ b_x k_x^2 \\ b_y k_y^2 \\ b_z k_z^2 \\ k_x^2 b_x^2 + k_y^2 b_y^2 + k_z^2 b_z^2 - g^2 \end{bmatrix}$$

Tìm giá trị đơn của $A \in R^{N \times 7}$

$$A = U \Sigma V'$$

Tham số p có thể tìm thấy từ

$$p = cV(:, 7)$$

Mà c được cho bởi

$$c \left(\frac{v(1, 7)^2}{v(1, 7)} + \frac{v(5, 7)^2}{v(2, 7)} + \frac{v(6, 7)^2}{v(3, 7)} - v(7, 7) \right) = g^2$$

Công cụ ước lượng UISL (gần đúng ellipsoid) : Để cho $x \in R^{N \times 1}, y \in R^{N \times 1}, z \in R^{N \times 1}$, tìm $k \in R^{3 \times 1}$ đáp ứng

$$\min_{k,b} \sum_{i=1}^N (k_1(x_i - b_1)^2 + k_2(y_i - b_2)^2 + k_3(y_i - b_3)^2 - r^2)^2$$

$$[k,b] = \text{ellipsoid1}(x,y,z,r)$$

Công cụ ước lượng UISL : tìm k và b từ y_a

Sử dụng tất cả y_a , để tìm k và b

$$[k,b] = \text{calibrationacc}(y_a)$$

Giả sử rằng y_a là hằng số, y_a dữ liệu trong suốt thời gian chuyển đổi (giữa 2 khoảng thời gian không đổi) bị bỏ qua.

$$[k,b] = \text{calibrationacc2}(y_a)$$

2.8. Các phương trình cảm biến

Ngõ ra cảm biến : ngõ ra vận tốc góc và gia tốc được tính theo công thức (2.13) [6]

$$\begin{aligned} y_g &= \omega_b + b_g + v_g \\ y_a &= a_b + C(q)g + b_a + v_a \end{aligned} \quad (2.13)$$

2.9. Ước lượng thay đổi và vị trí

Quaternion q tính toán từ phương trình (1.9). Cho \hat{q} được tính từ phương trình (2.14) [6]

$$\dot{\hat{q}} = \frac{1}{2}\dot{\Omega}(y_g)\hat{q} \quad (2.14)$$

Từ $y_g \neq \omega_b$ (y_g bao gồm nhiều cảm biến v_g và dự kiến vận tốc góc b_g), $\hat{q} \neq q$. Từ đó q_e được sử dụng để chỉ ra 1 lỗi nhỏ trong \hat{q} [6]

$$q = \hat{q} \otimes q_e \quad (2.15)$$

Mà q_e nhỏ và có thể xấp xỉ bằng:

$$q_e \approx \begin{bmatrix} 1 \\ \bar{q}_e \end{bmatrix} \quad (2.16)$$

2.9.1. Tính \hat{q}

Phương trình (2.14) gần đúng bởi một phương trình rời rạc:

$$\begin{aligned} \Omega^2 &= -\|\omega\|_2^2 I \\ \Omega^3 &= -\|\omega\|_2^2 \Omega \\ \dot{\Omega}_k &\approx \frac{\Omega_k - \Omega_{k-1}}{T} \end{aligned}$$

Gần đúng lần thứ 2:

$$\begin{aligned} q_{k+1} &\approx q_k + \frac{1}{2}\Omega T + \frac{1}{8}\Omega^2 T^2 + \frac{1}{4}\dot{\Omega} T^2 \\ &= q_k + \frac{1}{2}\Omega_k T - \frac{1}{8}\|\omega_k\|_2^2 T^2 + \frac{1}{4}\frac{\Omega_k - \Omega_{k-1}}{T} T^2 \\ &= \left(I + \frac{3}{4}\Omega_k T - \frac{1}{4}\Omega_{k-1} T - \frac{1}{8}\|\omega_k\|_2^2 T^2 \right) q_k \end{aligned}$$

Gần đúng lần thứ 3

$$\begin{aligned} q_{k+1} &= \left(I + \frac{3}{4}\Omega_k T - \frac{1}{4}\Omega_{k-1} T - \frac{1}{8}\|\omega_k\|_2^2 T^2 \right) q_k + \left(\frac{1}{48}\Omega^3 + \frac{1}{24}\Omega\dot{\Omega} \right) T^3 q_k \\ &= \left(I + \frac{3}{4}\Omega_k T - \frac{1}{4}\Omega_{k-1} T - \frac{1}{6}\|\omega_k\|_2^2 T^2 - \frac{1}{24}\Omega_k \Omega_{k-1} T^2 - \frac{1}{48}\|\omega_k\|_2^2 \Omega_k T^3 \right) q_k \end{aligned}$$

2.9.2 Phương trình \bar{q}_e

Với giả thiết của phương trình (2.16), chúng ta quan tâm với \bar{q}_e

$$\text{Phép luận I: } \dot{q}_e = \frac{1}{2}q_e \otimes \begin{bmatrix} 0 \\ (\omega_b - y_g) \end{bmatrix} + \begin{bmatrix} 0 \\ (\bar{q}_e \times y_g) \end{bmatrix} \quad (2.17)$$

Chứng minh :

$$\begin{aligned} \dot{q} &= \dot{\hat{q}} \otimes q_e + \hat{q} \otimes \dot{q}_e \\ \frac{1}{2}q \otimes \begin{bmatrix} 0 \\ \omega_b \end{bmatrix} &= \frac{1}{2}\hat{q} \otimes \begin{bmatrix} 0 \\ y_g \end{bmatrix} \otimes q_e + \hat{q} \otimes \dot{q}_e \end{aligned}$$

Vi phân \hat{q}^* 2 vế, ta có

$$\dot{q}_e = \frac{1}{2} q_e \otimes \begin{bmatrix} 0 \\ \omega_b \end{bmatrix} - \begin{bmatrix} 0 \\ y_g \end{bmatrix} \otimes q_e$$

Thực tế sử dụng

$$a \otimes b = b \otimes a + 2 \begin{bmatrix} 0 \\ \bar{a} \times \bar{b} \end{bmatrix}$$

Từ phương trình (1.17) ta có

Phép luận II : dẫn xuất của \bar{q}_e

$$\dot{\bar{q}}_e \approx -y_g \times \bar{q}_e - \frac{1}{2}(b_g + v_g) \quad (2.18)$$

Từ phương trình (2.17) ta có

$$\dot{\bar{q}}_e = \frac{1}{2}(\omega_b - y_g) + \frac{1}{2}\bar{q}_e \times (\omega_b - y_g) - y_g \times \bar{q}_e$$

2.9.3. Vận tốc và vị trí

Chúng ta sử dụng phương trình (2.2) và (2.3) vào phương trình (2.9) để tính v_n và r_n

$$\hat{v} = C'(\hat{q})y_a - \hat{g} \quad (2.19)$$

$$\hat{r} = \hat{v}$$

Từ $C'(q) \neq C'(\hat{q})y_a - g$, chúng ta có $v_n \neq \hat{v}$ và $\hat{r} \neq r$. v_e và r_e được sử dụng

Cho lỗi vận tốc và lỗi vị trí, tương ứng:

$$v_e \triangleq v_n - \hat{v}, r_e \triangleq r_n - \hat{r}$$

-Lỗi vận tốc

$$\begin{aligned} \dot{v}_e &= \dot{v} - \dot{\hat{v}} \\ &= C'(q)a_b - C'(\hat{q})y_a + \hat{g} \\ &= C'(q)(y_a - C(q)g - b_a - v_a) - C'(\hat{q})y_a + \hat{g} \\ &= (C'(q) - C'(\hat{q}))y_a - C'(q)(b_a + v_a) + \hat{g} - g \\ &= -(2K(\bar{q}_e)C(\hat{q}))'y_a - C'(\hat{q})(b_a + v_a) - (2K(\bar{q}_e)C(\hat{q}))'(b_a + v_a) + \hat{g} - g \\ &\approx -2C'(\hat{q})K(y_a)\bar{q}_e + C'(\hat{q})(-b_a - v_a) \end{aligned}$$

Mà chúng ta thường sử dụng

$$\begin{aligned} C(q_e) &\approx I - 2(\bar{q}_e \times) \\ C(q) &\approx C(\hat{q}) - 2K(\bar{q}_e)C(\hat{q}) \end{aligned} \quad (2.20)$$

-Lỗi vị trí được tính theo công thức (2.21) [6]

$$\dot{r}_e = \dot{r} - \dot{\hat{r}} = v - \hat{v} = v_e \quad (2.21)$$

2.10. AHRS

2.10.1 Bộ lọc cho AHRS

-Phương trình trạng thái x được viết theo phương trình (2.22) [6]

$$x = \begin{bmatrix} q_e \\ b_g \\ b_a \end{bmatrix} \in R^{9 \times 1} \quad (2.22)$$

- Phương trình trạng thái liên tục

$$\dot{x} = \begin{bmatrix} -K(y_g) & -0.5I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} -0.5v_g \\ v_{bg} \\ v_{ag} \end{bmatrix}$$

Mà

$$Q = \begin{bmatrix} 0.25R_g & 0 & 0 \\ 0 & Q_{bg} & 0 \\ 0 & 0 & Q_{ag} \end{bmatrix}$$

- Rời rạc hóa

$$x_{k+1} = \exp(AT) x_k + w_k$$

Mà Q_d được tính bởi

$$\begin{aligned} \int_0^T \exp(Ar) Q(r) \exp(Ar)' dr &\approx \int_0^T (I + Ar) Q(r) (I + Ar)' dr \\ &\approx \int_0^T Q(r) + ArQ(r) + Q(r)A' r dr \\ &\approx Q(0)T + \frac{T^2}{2} AQ(0) + \frac{T^2}{2} Q(0)A' \end{aligned}$$

- Phép đo

$$\begin{aligned} y_m &= C(q)\tilde{g} + a_b + v_a + b_a \\ y_m &= C(q)\tilde{m} + v_m \end{aligned}$$

Mà

$$\tilde{m} = \begin{bmatrix} \cos\alpha \\ 0 \\ -\sin\alpha \end{bmatrix}$$

-Sử dụng phương trình (2.20), ta có

$$\begin{aligned} y_a - C(\hat{q})\tilde{g} &= -2K(q_e)C(\hat{q})\tilde{g} + a_b + v_a + b_a \\ y_m - C(\hat{q})\tilde{m} &= -2K(q_e)C(\hat{q})\tilde{m} + v_m \end{aligned} \quad (2.23)$$

$$\begin{aligned} y_a - C(\hat{q})\tilde{g} &= 2K(C(\hat{q})\tilde{g})q_e + a_b + v_a + b_a \\ y_m - C(\hat{q})\tilde{m} &= 2K(C(\hat{q})\tilde{m})q_e + v_m \end{aligned} \quad (2.24)$$

- Cập nhật hai bước

+Cập nhật hấp dẫn đầu tiên

$$H_{a,k} \triangleq [2K(C(\hat{q})\tilde{g}) \quad 0 \quad I]$$

$$z_{a,k} \triangleq y_{a,k} - C(\hat{q}_k)\tilde{g}$$

$$K_{a,k} = P_k^- H'_{a,k} (H_{a,k} P_k^- H'_{a,k} + R_a)^{-1}$$

$$\begin{aligned}\hat{x}_{a,k} &= \hat{x}_k^- + K_{a,k}(z_{a,k} - H_{a,k}\hat{x}_k^-) \\ P_{a,k} &= (I - K_{a,k}H_{a,k})P_k^-\end{aligned}$$

+ Cập nhật cảm biến từ

$$H_{m,k} \triangleq [2K(C(\hat{q})\tilde{m}) \quad 0 \quad 0]$$

$$z_{m,k} \triangleq y_{m,k} - C(\hat{q}_k)\tilde{m}$$

$$P_{m,k}^- = \text{Diag}\{0, 0, P_{a,k}(3, 3), 0, 0, 0, 0, 0, 0, 0\}$$

$$K_{m,k} = P_{m,k}^- H_{m,k}' (H_{m,k} P_{m,k}^- H_{m,k}' + R_m)^{-1}$$

$$\hat{x}_k = \hat{x}_{a,k} + K_{m,k}(z_{m,k} - H_{m,k}\hat{x}_{a,k})$$

$$P_k = (I - K_{m,k}H_{m,k})P_{a,k}(I - K_{m,k}H_{m,k})' + K_{m,k}R_mK_{m,k}'$$

$$\hat{x}_{k+1}^- = \exp(AT)\hat{x}_k$$

$$P_{k+1}^- = \exp(AT)P_k\exp(AT)' + Q_d \quad (2.25)$$

- Phương pháp trực tiếp: Sử dụng phương trình (2.25) như một phương trình cảm biến

- Phương pháp gián tiếp: Giải quyết vấn đề tối ưu hóa sau

$$\min_{q_e} \lambda_1^2 \|y_a - C(\hat{q})\tilde{g} - 2K(C(\hat{q})\tilde{g})\|_2^2 + \lambda_2^2 \|y_m - C(\hat{q})\tilde{m} - 2K(C(\hat{q})\tilde{m})q_e\|_2^2 \quad (2.26)$$

$$\min_{q_e} \|Bq_e - c\|_2^2 \quad (2.27)$$

Mà

$$B = \begin{bmatrix} 2\lambda_1 K(C(\hat{q})\tilde{g}) \\ 2\lambda_2 K(C(\hat{q})\tilde{m}) \end{bmatrix}, C = \begin{bmatrix} \lambda_1(y_a - C(\hat{q})\tilde{g}) \\ \lambda_2(y_m - C(\hat{q})\tilde{m}) \end{bmatrix}$$

Cách giải quyết

$$q_e = (B'B)^{-1}B'c$$

Gián tiếp với gần đúng tối ưu

$$\min_{q_e} \|Bq_e - c\|_2^2 + \|\text{Diag}(\delta_1, \delta_2, \delta_3)q_e\|_2^2$$

$$q_e = (B'B + \text{Diag}(\delta_1^2, \delta_2^2, \delta_3^2))^{-1}B'c$$

- Gián tiếp 2: Sự tách biệt giữa (ϕ, θ) và ψ

+ Bước thứ nhất: tìm $q_e(1:2)$

$$\min_{q_e(1:2)} \left\| y_a - C(\hat{q})\tilde{g} - 2K(C(\hat{q})\tilde{g}) \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} q_e(1:2) \right\|_2^2$$

Tính toán \hat{q} mới: $\hat{q} \otimes q_e$ (mà $q_e(3) = 0$)

+ Bước thứ 2: Tìm $q_e(3)$

$$\min_{q_e(3)} \left\| y_m - C(\hat{q})\tilde{m} - 2K(C(\hat{q})\tilde{m}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} q_e(3) \right\|_2^2$$

2.11. Xác định góc quay từ giá trị đo

2.11.1 Thuật toán TRIAD

Giả sử véc tơ đơn vị a_i và b_i ($i=1,2$) được cho và chúng ta muốn tìm một ma trận quay đáp ứng C theo công thức (2.28) [6]

$$a_1 = Cb_1, a_2 = Cb_2 \quad (2.28)$$

Nếu a_i và b_i không là những véc tơ đơn vị, làm chúng thành những véc tơ đơn vị

Ba véc tơ trung gian

$$r_1 = b_1, r_2 = \frac{b_1 \times b_2}{\|b_1 \times b_2\|}, r_3 = r_1 \times r_2$$

$$s_1 = a_1, s_2 = \frac{a_1 \times a_2}{\|a_1 \times a_2\|}, s_3 = s_1 \times s_2$$

Bây giờ tồn tại một ma trận trực giao đáp ứng độc nhất C

$$Cr_i = s_i, i = 1, 2, 3$$

Ma trận quay C được tính bởi

$$C = s_1 r_1' + s_2 r_2' + s_3 r_3'$$

Trong thuật toán Triad, ma trận quay C được chọn sao cho $a_1 = Cb_1$ được thỏa mãn. Sau đó các thành phần không xác định trong C được xác định từ $a_2 = Cb_2$. Vì vậy phép quay từ thuật toán TRIAD tương tự như giải pháp của vấn đề tối ưu hóa với λ rất lớn:

$$\min_C \lambda \|a_1 - Cb_1\|_2^2 + \|a_2 - Cb_2\|_2^2 \quad (2.29)$$

Công cụ ước lượng UISL : Tính toán C sử dụng thuật toán TRIAD

Lệnh sử dụng trong Matlab: $[C] = \text{triad}(a,b)$

% a = (3 x 2)

% b = (3 x 2)

2.11.2. Xác định góc quay từ cảm biến

Đề cho y_a là đầu ra cảm biến gia tốc. Góc quay được tính toán sử dụng thuật toán TRIAD

$$y_a = C(q) \begin{bmatrix} 0 \\ 0 \\ 9.8 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = C(q) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Công cụ ước lượng UISL : Tính toán quaternion từ ngõ ra gia tốc

Lệnh sử dụng trong Matlab: $q = \text{quaternionya}(y_a)$

% ya : 3x1 accelerometer output

% q : quaternion

2.11.3. Thuật toán tổng quát

Tìm ma trận quay trong vấn đề tối ưu sau đây theo công thức (2.30) [6]

$$\min_C \frac{1}{2} \sum_{i=1}^n \lambda_i \|a_i - Cb_i\|_2^2 \quad (2.30)$$

Mà a_i và b_i là véc tơ đơn vị và λ_i là yếu tố trọng số. Khó khăn của C đó là $C \in SO(3)$: i.e., C phải là một ma trận trực giao với yếu tố quyết định 1

$L(c)$ được định nghĩa bởi

$$\begin{aligned}
 L(C) &\triangleq \frac{1}{2} \sum_{i=1}^n \lambda_i \|a_i - C b_i\|_2^2 \\
 L(C) &= \frac{1}{2} \sum_{i=1}^n \lambda_i (\|a_i\|_2^2 + \|C b_i\|_2^2 - 2a_i' C b_i) \\
 &= \frac{1}{2} \sum_{i=1}^n \lambda_i (1 + 1) - \sum_{i=1}^n \lambda_i a_i' C b_i \\
 &= \sum_{i=1}^n \lambda_i - \sum_{i=1}^n \lambda_i \text{Tr}(a_i' C b_i) \\
 &= \sum_{i=1}^n \lambda_i - \sum_{i=1}^n \lambda_i - \text{Tr}(C B_i a_i') \quad (* 1) \\
 &= \sum_{i=1}^n \lambda_i - \text{Tr}\left(C \sum_{i=1}^n \lambda_i b_i a_i'\right) \\
 &= \sum_{i=1}^n \lambda_i - \text{Tr}(C M')
 \end{aligned}$$

Với

$$M = \sum_{i=1}^n \lambda_i a_i b_i'$$

$$(* 1) \text{Tr}(AB) = \text{Tr}(BA)$$

Lưu ý : $\min L(C)$ tương đương với $\max \text{Tr}(C M')$

Phép luận 3: Xem xét vấn đề sau với $C \in SO(3)$:

$$\max_C \text{Tr} C M'$$

Giải pháp tối ưu được cho như sau:

$$C = U \text{diag}[1, 1, (\det U)(\det V)] V' \quad (2.31)$$

Mà ma trận trực giao U và V từ giá trị đơn lẻ của M :

$$M = U \text{Diag}[s_1, s_2, s_3] V'$$

Chứng minh:

$$\text{Tr}(C M') = \text{Tr}(C V \text{Diag}[s_1, s_2, s_3] U') = \text{Tr}(U' C V \text{Diag}[s_1, s_2, s_3])$$

Giải pháp tối ưu phải thỏa mãn

$$U' C V = \text{Diag}[1, 1, (\det U)(\det V)]$$

Từ đó ta có phương trình (2.31)

Công cụ ước lượng UISL : tính toán C từ a_i và b_i và λ_i

Lệnh sử dụng trong Matlab: $[C] = \text{dcmfromab}(a, b, \text{lambda})$
 % a = (3 x N) matrix
 % b = (3 x N) matrix
 % lambda = (1 x N) matrix

2.11.4. Khởi tạo góc quay ban đầu sử dụng cảm biến từ trường và cảm biến gia tốc

- Cảm biến gia tốc theo công thức (2.32) [6]

$$y_a = C \tilde{g} \quad (2.32)$$

- Cảm biến từ trường theo công thức (2.33) [6]

$$y_m = C \tilde{m} \quad (2.33)$$

Mà y_m được trung bình hóa (tức là $\|y_m\| = 1$)

- Góc nhúng α từ trường : ulsan (khoảng 50 độ)

+Vĩ độ : 35,34, kinh độ: 129.19

+Góc nhúng được tính bằng cách sử dụng chương trình USGS (United States Geological Survey's)

(<http://geomag.usgs.gov/>)

+Mô hình từ trường thế giới (toàn cầu) 2000.6.11 dữ liệu

$$\tilde{m} \triangleq \begin{bmatrix} m_{n,x} \\ 0 \\ m_{n,z} \end{bmatrix} = \begin{bmatrix} \cos\alpha \\ 0 \\ -\sin\alpha \end{bmatrix}$$

Sử dụng thuật toán TRIAD cho phương trình (2.26)

+Cho y_a và $y_m \Rightarrow$ tính trung bình y_a và y_m

$$\bar{y}_a = \frac{y_a}{\|y_a\|}, \quad \bar{y}_m = \frac{y_m}{\|y_m\|}$$

+Sử dụng TRIAD với

$$a_1 = \bar{y}_a, \quad b_1 = \frac{\tilde{g}}{\|\tilde{g}\|}, \quad a_2 = \bar{y}_m, \quad b_2 = \tilde{m}$$

+Sau đó ma trận quay được tính bởi

$$C = \begin{bmatrix} -\bar{y}_a \times \frac{\bar{y}_a \times \bar{y}_m}{\|\bar{y}_a \times \bar{y}_m\|} & \frac{\bar{y}_a \times \bar{y}_m}{\|\bar{y}_a \times \bar{y}_m\|} & \bar{y}_a \end{bmatrix} \quad (2.34)$$

-Công cụ ước lượng UISL: Tính toán C từ y_a và y_m sử dụng phương trình (2.34)

Lệnh sử dụng trong Matlab: $[C] = \text{dcmfromyaym}(y_a, y_m)$

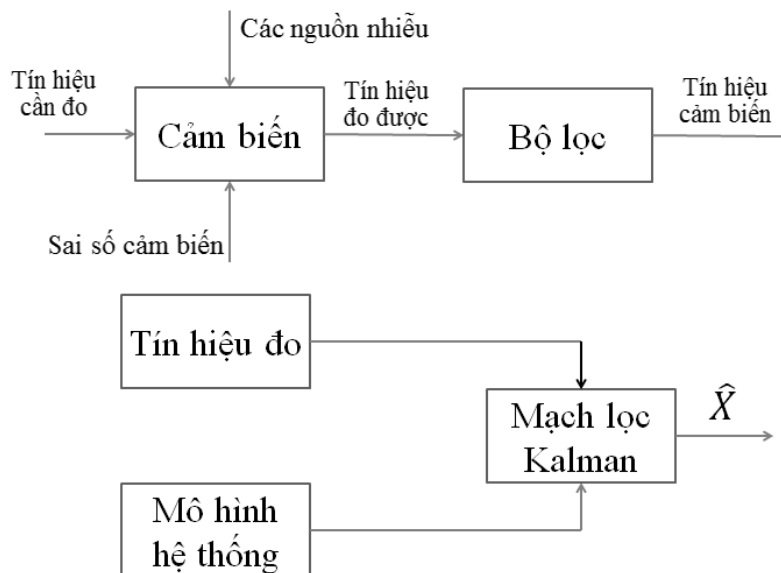
CHƯƠNG 3

BỘ LỌC KALMAN

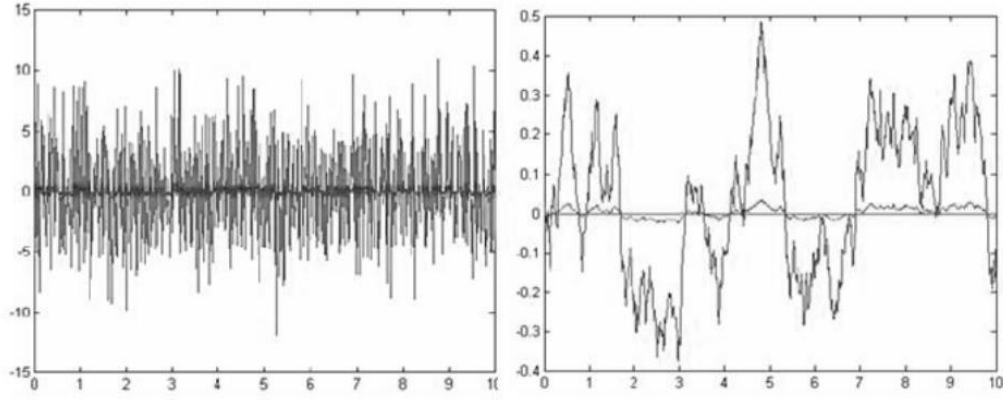
3.1 Giới thiệu chung về bộ lọc Kalman

Được đề xuất từ năm 1960 bởi giáo sư Kalman để thu thập và kết hợp linh động các thông tin từ cảm biến thành phần. Một khi phương trình định hướng và mẫu thống kê nhiều trên mỗi cảm biến được biết và xác định, bộ lọc Kalman sẽ cho ước lượng giá trị tối ưu (chính xác do đã được loại sai số, nhiễu) như là đang sử dụng một tín hiệu “tinh khiết” và có độ phân bố không đổi. Trong hệ thống này, tín hiệu cảm biến vào bộ lọc gồm hai tín hiệu: từ cảm biến góc (inclinometer) và cảm biến vận tốc góc (gyro). Tín hiệu đầu ra của bộ lọc là tín hiệu của inclinometer và gyro đã được loại nhiễu nhờ hai nguồn tín hiệu hỗ trợ và xử lý lẫn nhau trong bộ lọc, thông qua quan hệ (vận tốc góc = đạo hàm/vi phân của giá trị góc) [7].

Bộ lọc Kalman đơn giản là thuật toán xử lý dữ liệu hồi quy tối ưu. Có nhiều cách xác định tối ưu, phụ thuộc tiêu chuẩn lựa chọn trình thông số đánh giá. Nó cho thấy rằng bộ lọc Kalman tối ưu đối với chi tiết cụ thể trong bất kỳ tiêu chuẩn có nghĩa nào. Một khía cạnh của sự tối ưu này là bộ lọc Kalman hợp nhất tất cả thông tin được cung cấp tới nó. Nó xử lý tất cả giá trị sẵn có, ngoại trừ độ sai số, ước lượng giá trị hiện thời của những giá trị quan tâm, với cách sử dụng hiểu biết động học thiết bị giá trị và hệ thống, mô tả số liệu thống kê của hệ thống nhiễu, gồm nhiễu ồn, nhiễu đo và sự không chắc chắn trong mô hình động học, và những thông tin bất kỳ về điều kiện ban đầu của giá trị quan tâm [7].



Hình 3.1: Mô hình đo lường ước lượng của bộ lọc Kalman



Hình 3.2: Tín hiệu thu trước và sau khi lọc qua Kalman

Hình trên mô hình hóa hoạt động của mạch lọc Kalman. Chúng ta có tín hiệu đo được, chúng ta có mô hình của tín hiệu đo được (đòi hỏi tuyến tính) và sau đó là áp dụng vào trong hệ thống phương trình của mạch lọc để ước lượng trạng thái quan tâm. Thực ra tín hiệu đo là không khó, phương trình đã có sẵn, cái chúng ta cần chính là mô hình hoá hệ thống. Để có thể ứng dụng một cách hiệu quả mạch lọc Kalman thì chúng ta phải mô hình hóa được một cách tuyến tính sự thay đổi của trạng thái cần ước lượng hoặc dự đoán [7]

3.2. Mô hình toán học

3.2.1. Hệ thống và mô hình quan sát

Chúng ta giả sử rằng có thể mô hình hóa bởi phương trình chuyển trạng thái sau [7]:

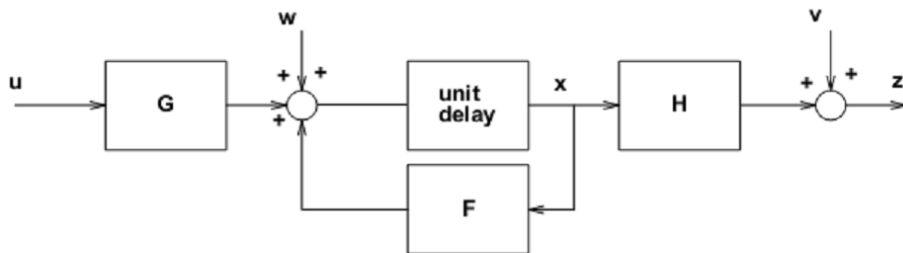
$$x_{k+1} = F_k x_k + G_k u_k + w_k$$

Trong đó x_k là trạng thái tại thời điểm k , u_k là vector điều khiển đầu vào, w_k là hệ thống cộng hay nhiễu quá trình – thường là nhiễu Gaussian trắng cộng (AWGN), G_k là ma trận chuyển đổi đầu vào và F_k là ma trận chuyển trạng thái.

Ngoài ra chúng ta giả sử rằng, khả năng quan sát trạng thái được thực hiện thông qua một hệ thống đo lường có thể được biểu diễn bởi một phương trình tuyến tính như sau [7]

$$z_k = H_k x_k + v_k$$

Trong đó z_k là thông tin quan sát hay đo lường thực hiện tại thời điểm k , x_k là trạng thái tại thời điểm k , H_k là ma trận quan sát và v_k là nhiễu cộng trong quá trình đo lường



Hình 3.3: Mô hình không gian trạng thái

3.2.2. Giả thiết

Chúng ta giả thiết như sau:

-Nhiều quá trình và nhiễu đo lường w_k và v_k là không tương quan, là nhiễu Gaussian trắng cộng (AWGN) có giá trị trung bình bằng không và ma trận hiệp phương sai đã biết.

Khi đó,

$$E[w_k w_l^T] = \begin{cases} Q_k & k = l \\ 0 & k \neq l \end{cases}$$

$$E[v_k v_l^T] = \begin{cases} R_k & k = l \\ 0 & k \neq l \end{cases}$$

$$E[w_k v_l^T] = 0 \quad \text{Cho tất cả } k, l$$

Trong đó Q_k và R_k là các ma trận đối xứng nửa xác định dương.

- Trạng thái khởi tạo hệ thống x_0 là một vector ngẫu nhiên không tương quan với cả hệ thống và nhiễu đo lường.

- Trạng thái khởi tạo hệ thống có giá trị trung bình và ma trận hiệp phương sai đã biết theo công thức (3.1) [7]

$$\hat{x}_{0|0} = E[x_0] \text{ và } P_{0|0} = E[(\hat{x}_{0|0} - x_0)(\hat{x}_{0|0} - x_0)^T] \quad (3.1)$$

Đưa ra những giả định trên với mục đích để xác định, đưa ra tập giá trị quan sát z_1, \dots, z_{k+1} , bộ lọc ước lượng ở thời điểm $+1$ tạo ra một ước lượng tối ưu của trạng thái x_{k+1} mà chúng ta ký hiệu bởi \hat{x}_{k+1} , tối thiểu hóa kỳ vọng của hàm tổn thất bình phương lỗi.

$$E[\|x_{k+1} - \hat{x}_{k+1}\|^2] = E[(x_{k+1} - \hat{x}_{k+1})^T (x_{k+1} - \hat{x}_{k+1})]$$

3.2.3. Nguồn gốc

Ký hiệu ước lượng dự đoán của trạng thái x_{k+1} , dựa trên quan sát ở thời điểm k , z_1, \dots, z_k là $\hat{x}_{k+1|k}$. Đó gọi là một bước trước dự đoán hay đơn giản là dự đoán. Bây giờ, giải pháp để tối thiểu hóa phương trình (3.1) là kỳ vọng của trạng thái ở thời điểm $k+1$ được ước định dựa trên quan sát ở thời điểm k . Như vậy,

$$\hat{x}_{k+1|k} = E[x_{k+1}|z_1, \dots, z_k] = E[x_{k+1}|Z^k]$$

Khi đó trạng thái dự đoán được cho bởi

$$\hat{x}_{k+1|k} = E[x_{k+1}|Z^k]$$

$$\hat{x}_{k+1|k} = E[F_k x_k + G_k u_k + w_k | Z^k]$$

$$\hat{x}_{k+1|k} = F_k E[x_k | Z^k] + G_k u_k + E[w_k | Z^k]$$

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k} + G_k u_k$$

Khi sử dụng trong thực tế, nhiễu quá trình có giá trị trung bình là 0 và u_k đã được biết chính xác.

Hiệp phương sai ước lượng dự đoán $P_{k+1|k}$ là trung bình bình phương sai số trong ước lượng $\hat{x}_{k+1|k}$.

Vì vậy, việc sử dụng các sự kiện mà w_k và $\hat{x}_{k+1|k}$ là không tương quan:

$$\begin{aligned} P_{k+1|k} &= cov(x_{k+1} - \hat{x}_{k+1|k}) \\ P_{k+1|k} &= cov(F_k(x_k - \hat{x}_{k|k}) + w_k) \\ P_{k+1|k} &= F_k cov(x_k - \hat{x}_{k|k}) F_k^T + E[w_k w_k^T] \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k \end{aligned} \quad (3.2)$$

Đang có một ước lượng dự đoán $\hat{x}_{k+1|k}$, giả sử rằng chúng ta đang có một giá trị quan sát z_{k+1} . Làm sao để sử dụng thông tin này để cập nhật trạng thái dự đoán, tức là tìm $\hat{x}_{k+1|k+1}$. Chúng ta giả sử rằng ước lượng là tổng trọng số tuyến tính của dự đoán và quan sát mới và có thể được mô tả bởi phương trình,

$$\hat{x}_{k+1|k+1} = K'_{k+1} \hat{x}_{k+1|k} + K_{k+1} z_{k+1} \quad (3.3)$$

Trong đó K'_{k+1} và K_{k+1} là những ma trận hiệu chỉnh bù hay ma trận độ lờ (của các kích thước khác nhau). Vấn đề của chúng ta bây giờ là tìm K'_{k+1} và K_{k+1} để tối thiểu hóa điều kiện ước lượng trung bình bình phương sai số. Sai số dự đoán được cho bởi

$$K'_{k+1|k+1} = \hat{x}_{k+1} - x_{k+1} \quad (3.4)$$

3.2.4. Điều kiện không chệch

Để điều kiện không chệch yêu cầu $E[\hat{x}_{k+1|k+1}] = E[x_{k+1}]$. Giả sử rằng $\hat{x}_{k|k}$ là một ước lượng không chệch. Kết hợp phương trình (3.3) và (3.4) và tính kỳ vọng theo công thức (3.5) [7]

$$\begin{aligned} E[\hat{x}_{k+1|k+1}] &= E[K'_{k+1} \hat{x}_{k+1|k} + K_{k+1} H_{k+1} x_{k+1} + K_{k+1} v_{k+1}] \\ &= K'_{k+1} E[\hat{x}_{k+1|k}] + K_{k+1} H_{k+1} E[x_{k+1}] + K_{k+1} E[v_{k+1}] \end{aligned} \quad (3.5)$$

Vì $E[v_{k+1}] = 0$, và dự đoán không chệch:

$$\begin{aligned} E[\hat{x}_{k+1|k}] &= E[F_k \hat{x}_{k|k} + G_k u_k] \\ E[\hat{x}_{k+1|k}] &= F_k E[\hat{x}_{k|k}] + G_k u_k \\ E[\hat{x}_{k+1|k}] &= E[x_{k+1}] \end{aligned} \quad (3.6)$$

Do đó kết hợp phương trình (3.5) và (3.6)

$$E[\hat{x}_{k+1|k+1}] = (K'_{k+1} + K_{k+1} H_{k+1}) E[x_{k+1}]$$

Và điều kiện để $\hat{x}_{k+1|k+1}$ không chệch yêu cầu

$$K'_{k+1} + K_{k+1} H_{k+1} = I$$

Hay $K'_{k+1} = I - K_{k+1} H_{k+1}$

Để ước lượng không chệch yêu cầu

$$\hat{x}_{k+1|k+1} = (I - K_{k+1} H_{k+1}) \hat{x}_{k+1|k} + K_{k+1} z_{k+1}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}[z_{k+1} - H_{k+1}z_{k+1}] \quad (3.7)$$

Trong đó K được gọi là độ lờ của bộ lọc Kalman (Kalman gain)

$H_{k+1}\hat{x}_{k+1|k}$ có thể được hiểu như một quan sát hay lường dự đoán $\hat{z}_{k+1|k}$

$$\hat{z}_{k+1|k} = E[z_{k+1}|Z^k]$$

$$\hat{z}_{k+1|k} = E[H_{k+1}x_{k+1} + v_{k+1}|Z^k]$$

$$\hat{z}_{k+1|k} = H_{k+1}\hat{x}_{k+1|k}$$

Đặt r_{k+1} là độ lệch đo lường thể hiện sự khác nhau giữa giá trị đo lường z_{k+1} và ước lượng của nó $\hat{z}_{k+1|k}$, được biểu diễn bởi

$$r_{k+1} = z_{k+1} - H_{k+1}\hat{x}_{k+1|k} \quad (3.8)$$

$$E[r_{k+1}|Z^k] = E[z_{k+1} - \hat{z}_{k+1|k}|Z^k]$$

$$E[r_{k+1}|Z^k] = H_{k+1}\hat{x}_{k+1|k} = 0$$

Kết hợp (3.7) và (3.8) ta được

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}r_{k+1}$$

Hiệp phương sai độ lệch đo lường S_{k+1} được cho bởi,

$$S_{k+1} = E[r_{k+1}r_{k+1}^T]$$

$$S_{k+1} = E[(z_{k+1} - H_{k+1}\hat{x}_{k+1|k})(z_{k+1} - H_{k+1}\hat{x}_{k+1|k})^T]$$

$$S_{k+1} = H_{k+1}P_{k+1}H_{k+1}^T + R_{k+1} \quad (3.9)$$

3.2.5. Hiệp phương sai sai số

Chúng ta xác định hiệp phương sai sai số dự đoán của phương trình (3.2). Bây giờ chúng ta tính toán hiệp phương sai sai số điều chỉnh.

$$P_{k+1|k+1} = E[(\hat{x}_{k+1|k+1})(\hat{x}_{k+1|k+1})^T|Z^k]$$

$$= E[(x_{k+1} - \hat{x}_{k+1|k+1})(x_{k+1} - \hat{x}_{k+1|k+1})^T]$$

$$= cov(x_{k+1} - \hat{x}_{k+1|k+1})$$

$$= cov\left(x_{k+1} - \left(\hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - H_{k+1}\hat{x}_{k+1|k})\right)\right)$$

$$= cov\left(x_{k+1} - \left(\hat{x}_{k+1|k} + K_{k+1}(H_{k+1}x_{k+1} + v_{k+1} - H_{k+1}\hat{x}_{k+1|k})\right)\right)$$

$$= cov\left((I - H_{k+1}K_{k+1})(x_{k+1} - \hat{x}_{k+1|k}) - K_{k+1}v_{k+1}\right)$$

$$= cov(I - K_{k+1}H_{k+1})(x_{k+1} - \hat{x}_{k+1|k}) + cov(K_{k+1}v_{k+1})$$

$$= (I - K_{k+1}H_{k+1})cov(x_{k+1} - \hat{x}_{k+1|k})(I - K_{k+1}H_{k+1})^T + K_{k+1}cov(v_{k+1})K_{k+1}^T$$

$$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k}(I - K_{k+1}H_{k+1})^T + K_{k+1}R_{k+1}K_{k+1}^T \quad (3.10)$$

Ta có thể tính toán $P_{k+1|k+1}$ theo cách khác như sau

$$P_{k+1|k+1} = cov(x_{k+1} - \hat{x}_{k+1|k+1})$$

$$\begin{aligned}
&= \text{cov}(x_{k+1} - \hat{x}_{k+1} - K_{k+1}r_{k+1}) \\
&= \text{cov}(x_{k+1} - \hat{x}_{k+1}) - \text{cov}(K_{k+1}r_{k+1}) \\
&= P_{k+1|k} - K_{k+1}S_{k+1}K_{k+1}^T
\end{aligned} \tag{3.11}$$

Trong đó, $P_{k+1|k} = \text{cov}(x_{k+1} - \hat{x}_{k+1|k})$

$$R_{k+1} = \text{cov}(v_{k+1}) = E[v_{k+1}v_{k+1}^T]$$

Vì vậy hiệp phương sai của ước lượng điều chỉnh đã được biểu diễn qua hiệp phương sai dự đoán $P_{k+1|k}$ nhiễu đo lường R_{k+1} và ma trận độ lờ Kalman

3.2.6. Độ lờ Kalman

Mục tiêu của chúng ta là làm sao để tối thiểu hóa trung bình bình phương sai số ước lượng có điều kiện với độ lờ Kalman K theo [7]

$$\begin{aligned}
L &= \min_{K_{k+1}} E \left[(\tilde{x}_{k+1|k})^T (\tilde{x}_{k+1|k+1}) | Z^k \right] \\
&= \min_{K_{k+1}} \text{trace} \left(E \left[(\tilde{x}_{k+1|k+1})^T (\tilde{x}_{k+1|k+1}) | Z^k \right] \right) \\
&= \min_{K_{k+1}} \text{trace} (P_{k+1|k+1})
\end{aligned} \tag{3.12}$$

Với bất kỳ ma trận A và ma trận đối xứng B ta có

$$\frac{\partial}{\partial A} (\text{trace}(ABA^T)) = 2AB$$

Kết hợp phương trình (3.11) và (3.12) và lấy vi phân ma trận độ lờ và đặt kết quả bằng 0 ta được

$$\frac{\partial L}{\partial K_{k+1}} = -2(I - K_{k+1}H_{k+1})P_{k+1|k}H_{k+1}^T + 2K_{k+1}R_{k+1} = 0$$

Sắp xếp lại và đưa ra phương trình cho ma trận độ lờ

$$K_{k+1} = P_{k+1|k}H_{k+1}^T [H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}]^{-1}$$

Kết hợp với (3.9) ta được

$$K_{k+1} = P_{k+1|k}H_{k+1}^T S_{k+1}^{-1} \tag{3.13}$$

Cùng với phương trình (3.13), định nghĩa một ước lượng tối ưu tuyến tính trung bình bình phương sai số.

Từ phương trình (3.13) ta có

$$P_{k+1}S_{k+1}H_{k+1}^T = P_{k+1|k}H_{k+1}^T K_{k+1}^T \tag{3.14}$$

Kết hợp phương trình (3.11) và (3.14) ta có

$$\begin{aligned}
P_{k+1|k+1} &= P_{k+1|k}(I - H_{k+1}^T K_{k+1}^T) \\
&= \left(P_{k+1|k}(I - H_{k+1}^T K_{k+1}^T) \right)^T \\
&= (I - H_{k+1}^T K_{k+1}^T)^T (P_{k+1|k})^T
\end{aligned}$$

$$= (I - K_{k+1}H_{k+1})P_{k+1|k} \quad (3.15)$$

3.2.7. Tóm tắt các phương trình của bộ lọc Kalman

Trong phần này chúng ta sẽ tóm tắt các phương trình tổng quát của giải thuật lọc Kalman. Giải thuật bao gồm 2 quá trình: quá trình ước lượng và quá trình điều chỉnh.

- **Quá trình dự đoán [7]**

Bộ lọc Kalman dựa vào trạng thái ước lượng điều chỉnh $\hat{x}_{k|k}$ - là ước lượng của x_k để ước lượng trạng thái $\hat{x}_{k+1|k}$ - là ước lượng dự đoán của x_{k+1} cho phép đo z_{k+1}

Trạng thái dự đoán:

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k} + G_k u_k \quad (3.16)$$

Hiệp phương sai ước lượng dự đoán:

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \quad (3.17)$$

Đo lường dự đoán:

$$\hat{z}_{k+1|k} = H_{k+1} \hat{x}_{k+1|k} \quad (3.18)$$

- **Quá trình điều chỉnh [7]**

Đo độ lệch đo lường:

$$r_{k+1} = z_{k+1} - H_{k+1} \hat{x}_{k+1|k} \quad (3.19)$$

Hiệp phương sai độ lệch:

$$S_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \quad (3.20)$$

Độ lờn Kalman:

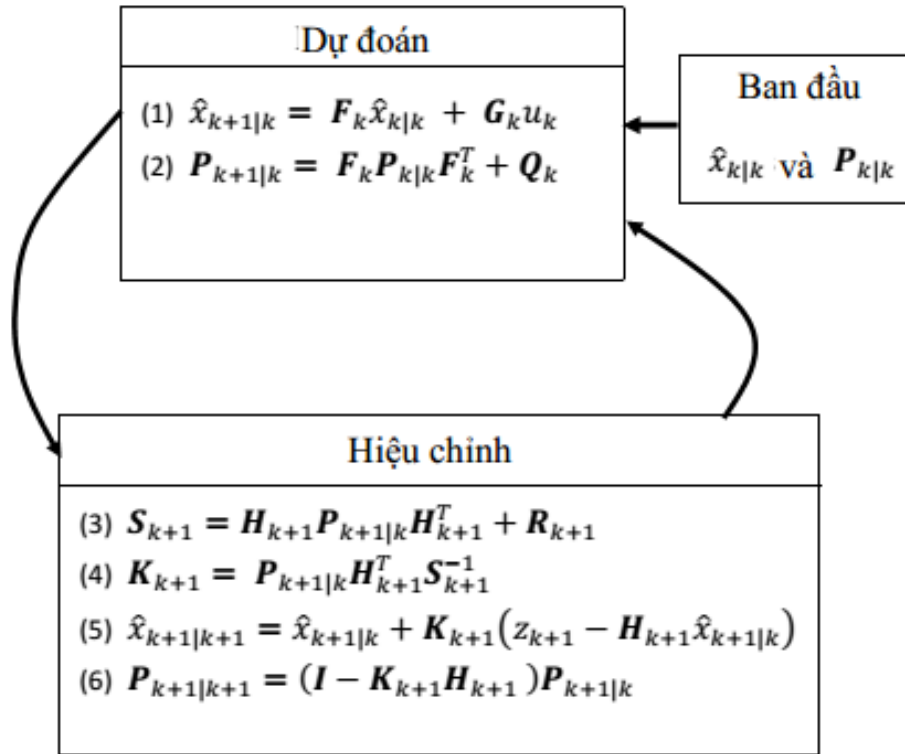
$$K_{k+1} = P_{k+1} H_{k+1}^T S_{k+1}^{-1} \quad (3.21)$$

Trạng thái ước lượng hiệu chỉnh:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} r_{k+1} \quad (3.22)$$

Hiệp phương sai ước lượng hiệu chỉnh:

$$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k} \quad (3.23)$$



Hình 3.4: Tóm tắt quá trình khởi tạo của Kalman

Cùng với các điều kiện ban đầu trong ước lượng và ma trận hiệp phương sai lỗi của nó (phương trình (3.1)) đã định nghĩa một giải thuật rời rạc hóa về thời gian và đệ quy để xác định hiệp phương sai ước lượng tuyến tính tối thiểu được gọi là bộ lọc Kalman [7].

CHƯƠNG 4

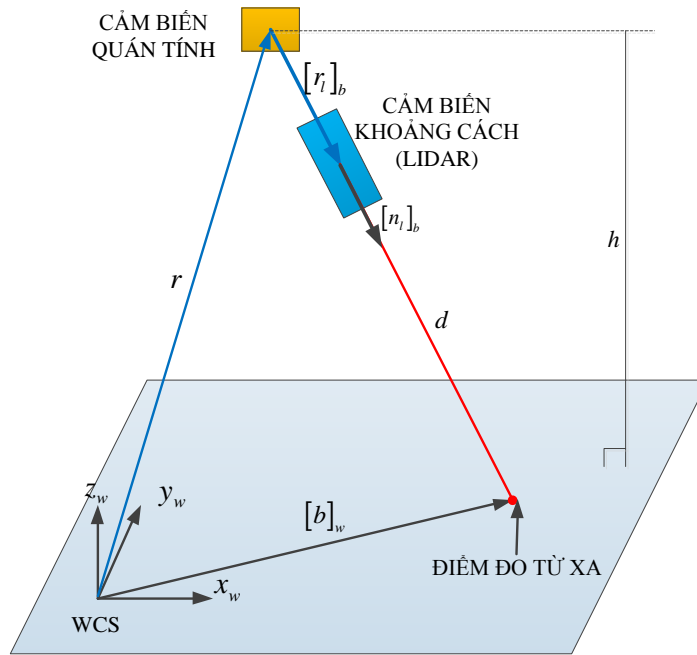
THUẬT TOÁN XÁC ĐỊNH THÔNG SỐ CỦA CẢM BIẾN VÀ THUẬT TOÁN ĐỊNH VỊ QUẢN TÍNH CHO BỘ LỌC KALMAN

4.1. Thuật toán xác định các thông số của cảm biến khoảng cách

Trong phần này, chúng tôi trình bày thuật toán nhằm xác định thông số vị trí $[r_l]_b$ và hướng $[n_l]_b$ của cảm biến khoảng cách đối với cảm biến quán tính. Trong đó, hệ thống được cầm trên tay trong khi cảm biến khoảng cách chỉ xuống sàn nhà. Tọa độ của điểm trên sàn nhà mà hệ thống đang chỉ vào trong hệ tọa độ WCS được tính theo công thức:

$$[b]_w = C_b^w [b]_b + r = C_b^w ([r_l] + d[n_l]_b) + r \quad (4.1)$$

Trong đó: r là vị trí của cảm biến quán tính trong hệ tọa độ WCS, C_b^w là ma trận quay từ BCS sang WCS, $[b]_b = [r_l] + d[n_l]_b$ là vị trí trong BCS của điểm trên sàn nhà mà hệ thống đang chỉ vào và d là khoảng cách từ hệ thống đến điểm đó và được xác định bằng cảm biến khoảng cách. r và C_b^w được ước lượng bằng thuật toán định vị quán tính (INA – Inertial Navigation Algorithm)



Hình 4.1 : Phương pháp xác định các thông số của cảm biến khoảng cách.

Tác giả cố gắng bố trí cảm biến khoảng cách và cảm biến quán tính sao cho $[r_l]_b = k[n_l]_b$ (Hình 1). Trong đó, $[n_l]_b$ là vector đơn vị và k là khoảng cách giữa cảm biến khoảng cách và cảm biến quán tính. Do k thường rất nhỏ (khoảng 3 cm) so với d (từ 3 ~ 40 m), nên k không ảnh hưởng lớn đến kết quả phép đo và được đo bằng thước.

Phương pháp ước lượng $[n_l]_b$ được mô tả trong Hình 4.1. Trong đó, cảm biến khoảng cách luôn chỉ xuống sàn nhà ở các góc nghiêng khác nhau và khoảng cách từ hệ thống xuống sàn nhà được đo bằng thước.

Do $[r_l] = k[n_l]_b$ nên công thức (4.1) trở thành

$$[b]_w = C_b^w (k + d)[n_l]_b + r \quad (4.2)$$

Do mặt sàn là phẳng và nằm ngang, nên độ cao của hệ thống được xác định bằng công thức (thành phần theo trục z)

$$[0 \ 0 \ 1](k + d)C_b^w[n_l]_b = -h \quad (4.3)$$

Trong đó h là độ cao của hệ thống và được đo bằng thước.

Nếu chúng ta lặp lại việc đo này n lần với các góc nghiêng khác nhau, ta có

$$\begin{bmatrix} [0 \ 0 \ 1](k + d_1)C_{1,b}^w \\ [0 \ 0 \ 1](k + d_2)C_{2,b}^w \\ \vdots \\ [0 \ 0 \ 1](k + d_n)C_{n,b}^w \end{bmatrix} [n_l]_b = \begin{bmatrix} -h_1 \\ -h_2 \\ \vdots \\ -h_n \end{bmatrix} \quad (4.4)$$

$C_{i,b}^w, i = 1, 2, \dots, m$ là ma trận quay từ BCS sang WCS tại thời điểm i được ước lượng từ thuật toán định vị quán tính

Như vậy, $[n_l]_b$ được ước lượng bằng việc tối thiểu sai số

$$\|S_1[n_l]_b - S_2\|_2^2 \quad (4.5)$$

Với

$$S_1 = \begin{bmatrix} [0 \ 0 \ 1](k + d_1)C_{1,b}^w \\ [0 \ 0 \ 1](k + d_2)C_{2,b}^w \\ \vdots \\ [0 \ 0 \ 1](k + d_n)C_{n,b}^w \end{bmatrix} \text{ và } S_2 = \begin{bmatrix} -h_1 \\ -h_2 \\ \vdots \\ -h_n \end{bmatrix}$$

Như vậy, tham số $[n_l]_b$ được tối ưu bởi công thức

$$[n_l]_b = (S_1^T S_1)^{-1} S_1^T S_2 \quad (4.6)$$

4.2. Thuật toán định vị quán tính (ina) sử dụng bộ lọc kalman

Trong phần này, tác giả trình bày về INA và việc ứng dụng bộ lọc Kalman vào bài toán định vị quán tính để xác định các tham số C_b^w và r trong công thức (4.1).

Đặt $v \in R^3$ và $r \in R^3$ là vận tốc và vị trí của cảm biến quán tính trong WCS. Đặt $C_w^b(q) \in R^{3 \times 3}$ là ma trận quay từ WCS sang BCS tương ứng với quaternion $q \in R^4$

Quaternion q , vận tốc v và vị trí r của thiết bị liên quan với nhau qua công thức :

$$\begin{aligned} \dot{q} &= \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} q \\ \dot{v} &= C_w^b(q)^T [a]_b \\ \dot{r} &= v \end{aligned} \quad (4.7)$$

Trong đó ω là vận tốc góc của BCS trong WCS và $[a]_b \in R^3$ là gia tốc tịnh tiến trong BCS. Giá trị đầu ra của cảm biến vận tốc góc ($y_g \in R^3$) và cảm biến gia tốc ($y_a \in R^3$) được cho bởi công thức:

$$\begin{aligned} y_g &= \omega + v_g + b_g \\ y_a &= [a]_b + C_w^b(q)[\tilde{g}]_w + v_a + b_a \end{aligned} \quad (4.8)$$

Trong đó $[\tilde{g}]_w \in R^3$ là vector gia tốc trọng trường trong WCS. $b_g \in R^3$ và $b_a \in R^3$ là thành phần nhiễu thay đổi chậm của cảm biến vận tốc góc và cảm biến gia tốc. v_g và v_a là thành phần nhiễu trắng của cảm biến vận tốc góc và cảm biến gia tốc.

Thuật toán tích phân để lấy tích phân công thức (4.7) (thay $[a]_b$ bằng $y_a - C_w^b(q)\tilde{g}$ và thay ω bằng y_g). Đặt \hat{q} , \hat{r} và \hat{v} là giá trị tích phân của quaternion, vị trí và vận tốc của cảm biến quán tính.

Do các cảm biến luôn có thành phần nhiễu, do vậy các giá trị tích phân đó chưa phải là giá trị đúng. Gọi $\bar{q} \in R^3$, $\bar{r} \in R^3$ và $\bar{v} \in R^3$ là sai số của quaternion, vị trí và vận tốc của cảm biến quán tính:

$$\begin{aligned}\bar{q} &= [0_{3 \times 1} \quad I_3] (\hat{q}^* \oplus q) \\ \bar{r} &= r - \hat{r} \\ \bar{v} &= v - \hat{v}\end{aligned}\tag{4.9}$$

Trong đó \oplus là phép nhân quaternion và q^* là quaternion liên hợp của q . Phương trình (4.9) biểu diễn 3 thành phần của sai số quaternion $\bar{q} \in R^3$

Các biến trạng thái được sử dụng trong bộ lọc Kalman:

$$x = \begin{bmatrix} \bar{q} \\ b_g \\ \bar{r} \\ \bar{v} \\ b_a \end{bmatrix} \in R^{15}\tag{4.10}$$

Phương trình trạng thái cho bộ lọc Kalman

$$\dot{x}(t) = A(t)x(t) + \omega(t)\tag{4.11}$$

Trong đó:

$$A(t) = \begin{bmatrix} [-y_g \times] & -\frac{1}{2}I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ -2C_w^b(\hat{q})^T[y_a \times] & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$w(t) = \begin{bmatrix} -\frac{1}{2}v_g \\ w_{bg} \\ 0 \\ -C_w^b(\hat{q})^T v_a \\ w_{ba} \end{bmatrix}$$

$[a \times] \in R^{3 \times 3}$ là ma trận đối xứng lệch tương ứng với vector $a \in R^{3 \times 1}$. Nhiễu w_{bg} và w_{ba} đại diện cho sự thay đổi nhỏ của thành phần nhiễu chậm tương ứng.

Trong quá trình sử dụng hệ thống, có những thời điểm vận tốc của hệ thống bằng 0 (ZVI – zero velocity interval). Chúng ta có thể sử dụng các ZVI này để cập nhật lại sai số cho giá trị vận tốc trong INA. Các ZVI này có thể được phát hiện trực tiếp bởi cảm biến vận tốc Doppler. Tuy nhiên, chúng ta có thể phát hiện các ZVI gián tiếp bằng cách sử dụng thuật toán phát hiện vận tốc bằng 0

Trong phần trình bày này, tác giả sử dụng một thuật toán phát hiện ZVI đơn giản. Nếu những điều kiện dưới đây được thỏa mãn thì thời điểm gián đoạn m phải thuộc ZVI:

$$\begin{aligned} \|y_{g,i}\| &\leq B_g, & m - \frac{N_g}{2} \leq i \leq m + \frac{N_g}{2} \\ \|y_{a,i} - y_{a,i-1}\| &\leq B_a, & m - \frac{N_a}{2} \leq i \leq m + \frac{N_a}{2} \end{aligned} \quad (4.12)$$

Trong đó N_g và N_a là các số nguyên. B_g, B_a là các giá trị đặt ngưỡng.

Trong khoảng ZVI này, ta có phương trình cập nhật vận tốc bằng 0:

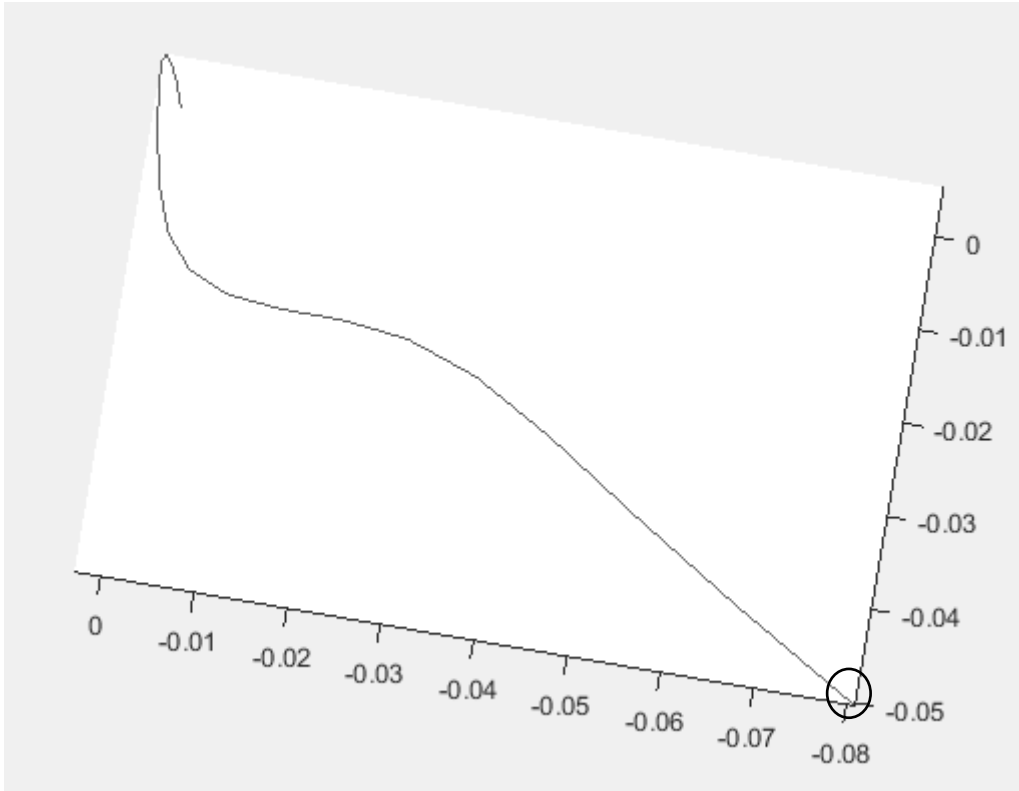
$$z_v = H_v x + v_{zero} \quad (4.13)$$

Với:

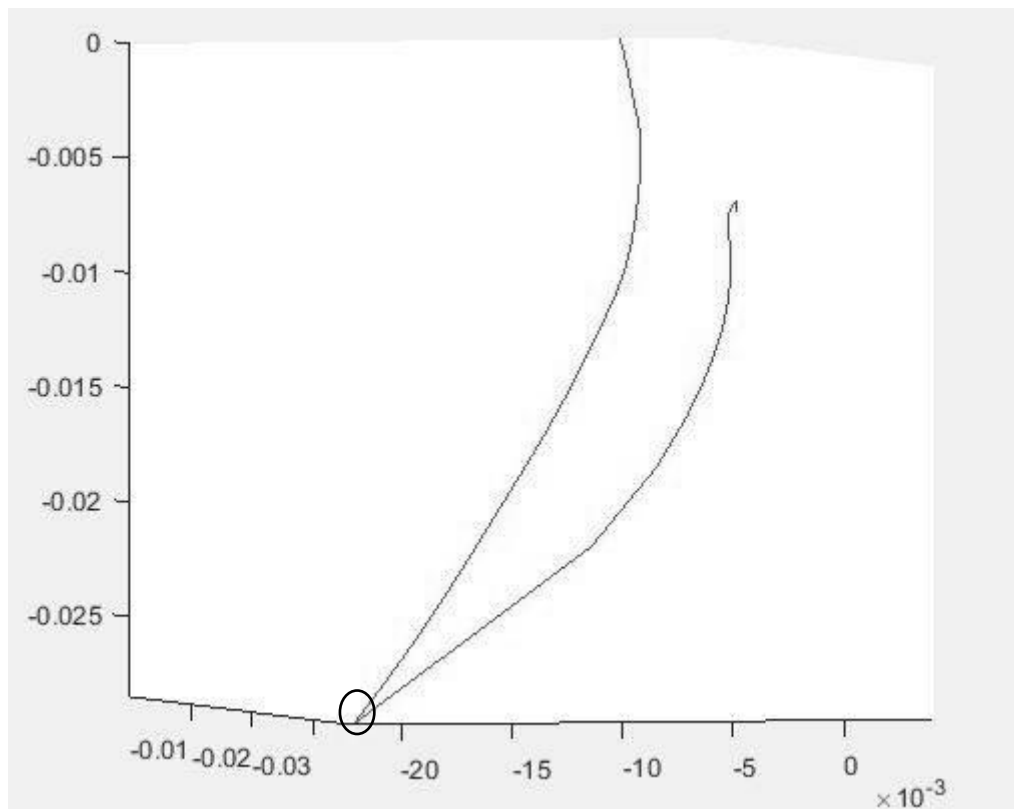
$$\begin{aligned} z_v &= 0_{3 \times 1} - \hat{v} \in R^{3 \times 1} \\ H_v &= [0_{3 \times 9} \quad I_3 \quad 0_{3 \times 3}] \end{aligned}$$

4.3. Ứng dụng bộ lọc Kalman cho thuật toán định vị quán tính

Để ứng dụng bộ lọc Kalman cho thuật toán định vị quán tính trong việc xác định vị trí của cảm biến, tác giả sử dụng bộ lọc Kalman cho phương trình trạng thái Kalman (4.11) để ước lượng vị trí của véc tơ vị trí \bar{r} khi không sử dụng bộ lọc Kalman và khi có sử dụng bộ lọc Kalman.



Hình 4.2: Véc tơ vị trí \bar{r} khi không sử dụng bộ lọc Kalman



Hình 4.3: Vec tơ vị trí \bar{r} khi có sử dụng bộ lọc Kalman

Giải thích biểu đồ hình 4.2 và 4.3:

Trong quá trình lấy dữ liệu người dùng sẽ thực hiện các bước nhấn nút để chỉ vào điểm thứ nhất, sau đó nhấn nút chỉ vào điểm thứ 2. Vì vậy quá trình lấy dữ liệu sẽ theo trình tự “dừng – chuyển động – dừng “. Vòng tròn trên hình 4.2 và 4.3 là điểm bắt đầu của chuyển động, điểm còn lại là điểm cuối chuyển động. Ta thấy hình 4.2 chỉ có 2 điểm, tương ứng với điểm bắt đầu và điểm cuối.

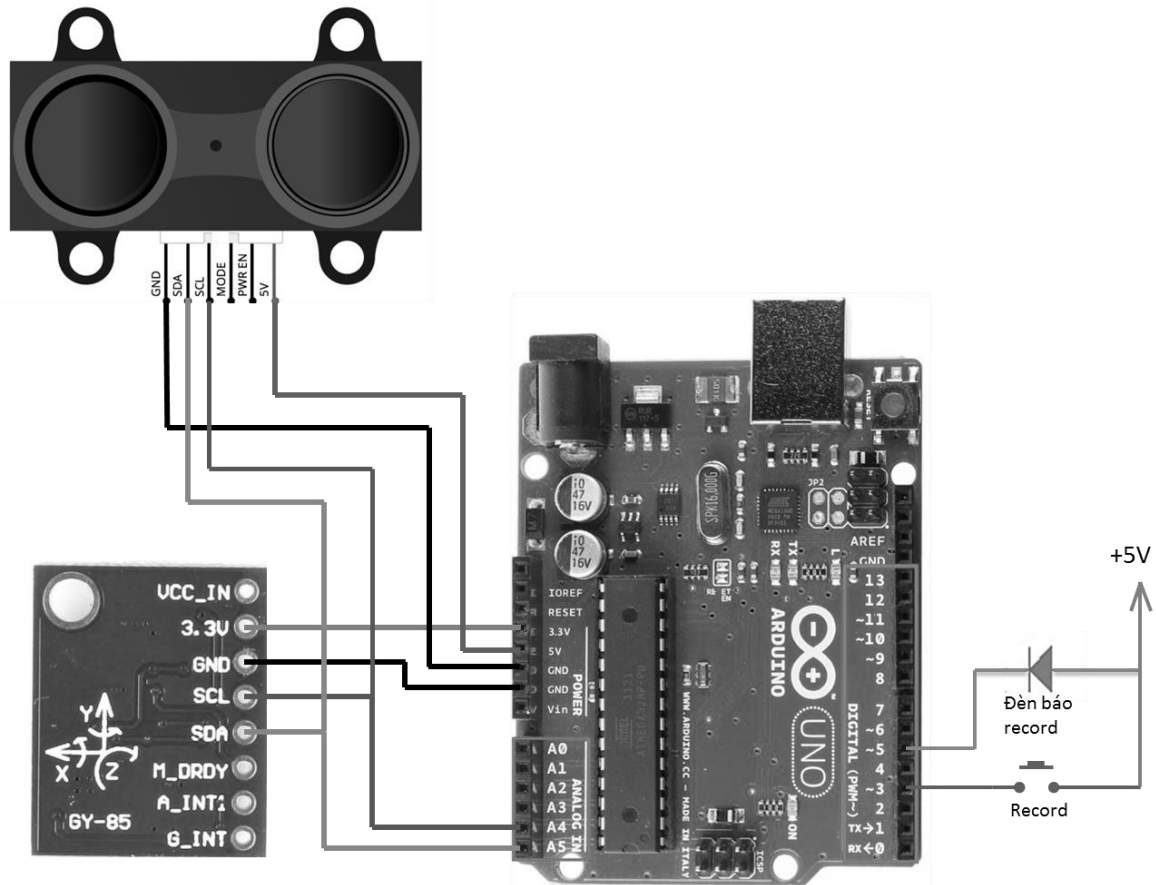
Giải thích biểu đồ hình 4.2 có 1 điểm đầu và điểm cuối do cảm biến luôn có nhiễu, mà phép định vị quán tính là kết hợp tích phân 2 lớp của gia tốc và tích phân 1 lớp của vận tốc góc. Khi tích phân sai số sẽ cộng dồn lên dẫn đến sai số lớn.

Kết luận : Việc sử dụng bộ lọc Kalman có hiệu quả trong việc ước lượng vị trí và giảm sai số một cách đáng kể.

CHƯƠNG 5

THIẾT KẾ PHẦN CỨNG, THỰC NGHIỆM ĐO KHOẢNG CÁCH VÀ ĐÁNH GIÁ KẾT QUẢ ĐO

5.1. Sơ đồ chi tiết



Hình 5.1: Sơ đồ thiết bị đo khoảng cách từ xa

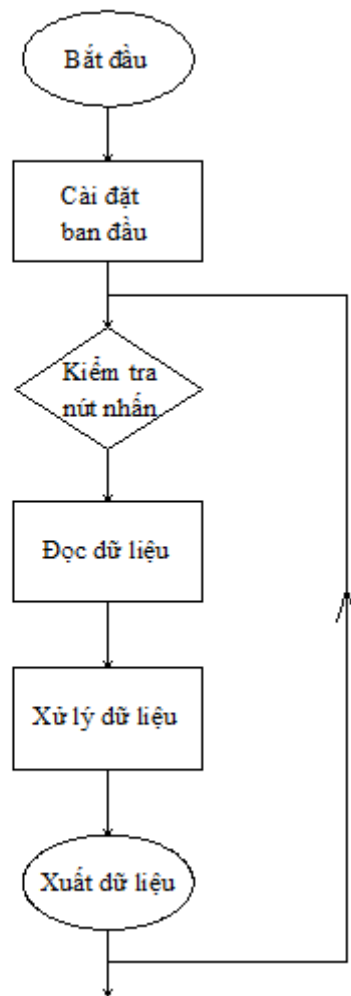
Giao tiếp giữa LIDAR với board vi điều khiển Arduino Uno R3 bằng bus I2C

Giao tiếp giữa IMU với vi điều khiển board Arduino Uno R3 bằng bus I2C

5.2. Đọc dữ liệu từ LIDAR

Giao tiếp giữa LIDAR với board vi điều khiển Arduino Uno R3 bằng bus I2C, khi hoạt động LIDAR truyền dữ liệu liên tục với tần số tối đa 100Hz, board vi điều khiển Arduino sẽ đọc dữ liệu 0x62 của LIDAR. Với giá trị 1 ms/m ta tính được khoảng cách từ máy đo đến điểm cần đo.

Lưu đồ thuật toán như sau:



Hình 5.2 : Lưu đồ thuật toán đọc dữ liệu từ LIDAR

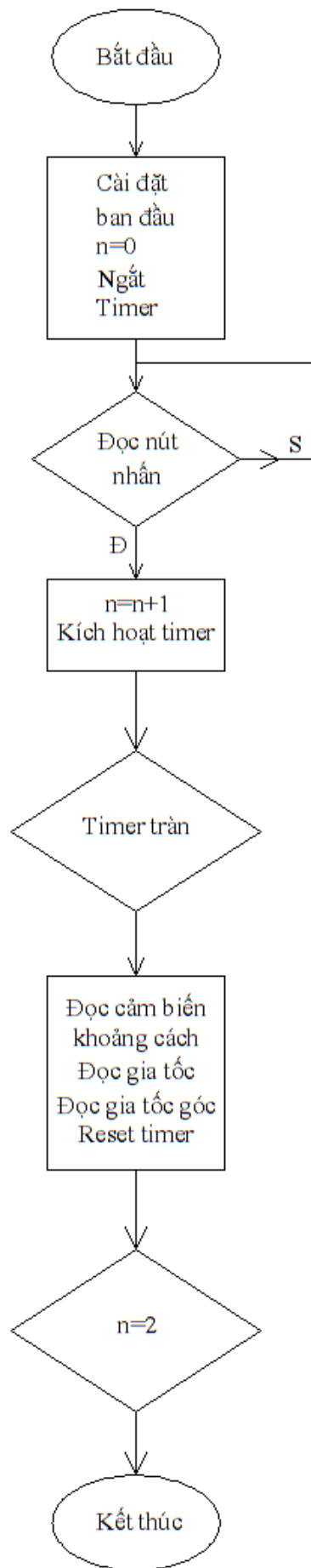
Giải thích lưu đồ thuật toán: Cài đặt ban đầu khoảng cách bằng 0, vi điều khiển sẽ quét liên tục để kiểm tra người dùng có nhấn nút hay không, khi có nhấn nút thì vi điều khiển sẽ đọc thời gian phát xung và nhận xung, từ đó tính toán được khoảng cách từ điểm cần đo đến thiết bị

5.3. Đọc dữ liệu từ GY-85

Tần số truyền dữ liệu của cảm biến đo xa LIDAR là 33Hz, tần số truyền dữ liệu của cảm biến quán tính GY-85 là 100Hz, do vậy để việc đo chính xác thì việc đo khoảng cách thì cảm biến khoảng cách phải đồng bộ với cảm biến quán tính. Việc đọc dữ liệu phải được thực hiện cùng thời điểm.

Giao tiếp giữa IMU GY-85 với board vi điều khiển Arduino UNO R3 bằng bus I2C, ban đầu đèn laser được bật lên, người dùng chỉ vào điểm cần đo thứ nhất, sau đó nhấn nút record, tiếp theo người dùng chỉ vào điểm cần đo thứ hai, sau đó nhấn nút record để kết thúc việc đọc dữ liệu. Dữ liệu được vi điều khiển đọc sau đó truyền qua giao diện giám sát nối tiếp (Serial monitor)

Lưu đồ thuật toán như sau:



Hình 5.3: Lưu đồ thuật toán chương trình đọc dữ từ GY-85

Giải thích lưu đồ thuật toán: Cài đặt ban đầu timer tần số 33.3Hz, ngắt tràn timer, $n=0$. Khi kích hoạt nút nhấn $n=1$, timer bắt đầu đếm, khi timer tràn thì ngắt xảy ra thì vi điều khiển sẽ đọc dữ liệu cảm biến khoảng cách, đọc dữ liệu của cảm biến quán tính, khi ngắt lập tức timer bị reset, khi timer bị reset thì tiếp tục đếm. Vòng lặp tiếp tục thực hiện khi người dùng nhấn nút lần 2 ($n=2$) thì thoát chương trình.

5.4. Lập trình cho vi điều khiển

Trước khi lập trình cho board vi điều khiển ARDUINO UNO R3, ta tìm và thêm các thư viện GY_85.h, I2C.h, UART_ARDUINO.h vào phần mềm Arduino.

Phần code chương trình được trình bày trong phụ lục 1

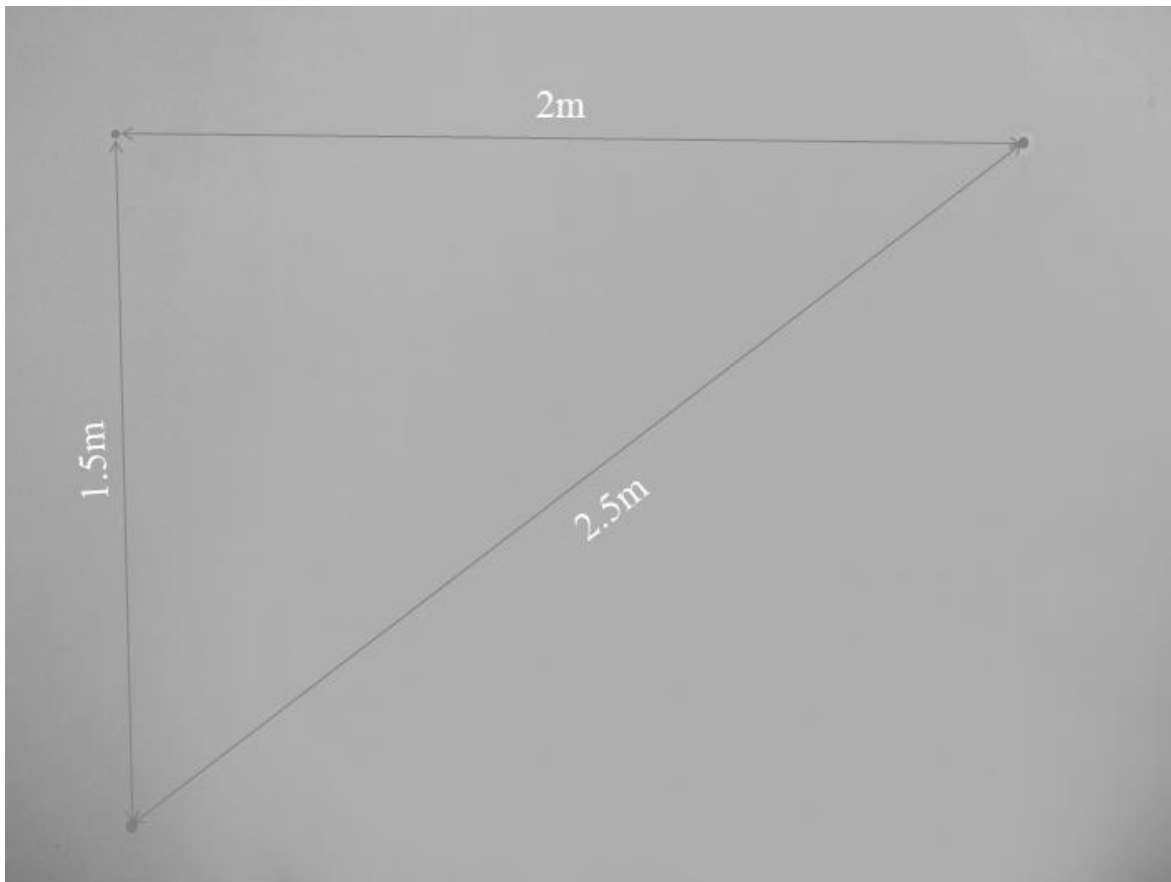
5.5. Viết chương trình cho phần mềm Matlab

Khi có được dữ liệu từ vi điều khiển, dữ liệu có dạng tập tin có phần mở rộng kiểu .mat

Ta viết code cho phần mềm Matlab được trình bày trong phụ lục 2

5.6. Cài đặt điểm đo

Để tiến hành thực nghiệm đo độ dài từ xa, tác giả thiết lập một bức tường trong một căn phòng, trên tường là một tam giác cố độ dài thực tế các cạnh $AB=2m$; $AC=1.5m$; $BC=2.5m$ như hình 7.1.

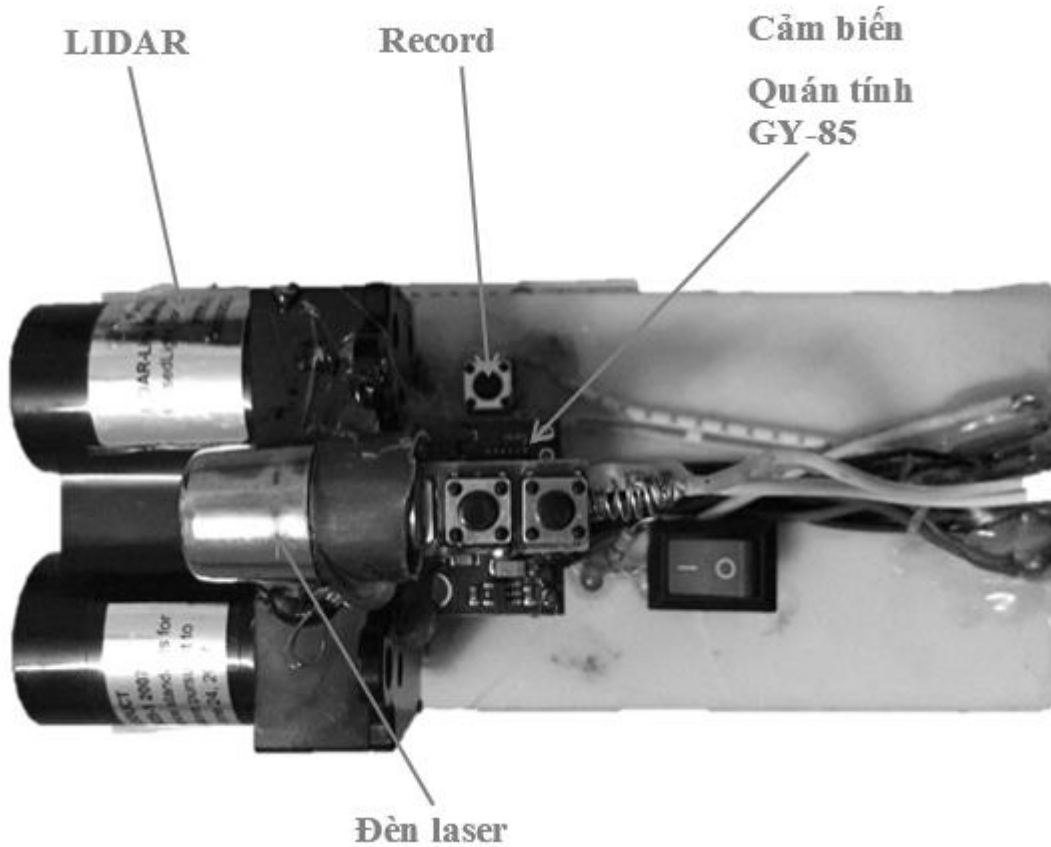


Hình 5.4: Các điểm cần đo trên bức tường

Khoảng cách từ thiết bị đo xa nhỏ hơn điểm cần đo 40m, người dùng có thể đứng thẳng hoặc chéo tùy ý so với phương vuông góc với bức tường. Như vậy các độ dài thực nghiệm cần đo là các đoạn AB, BC, AC.

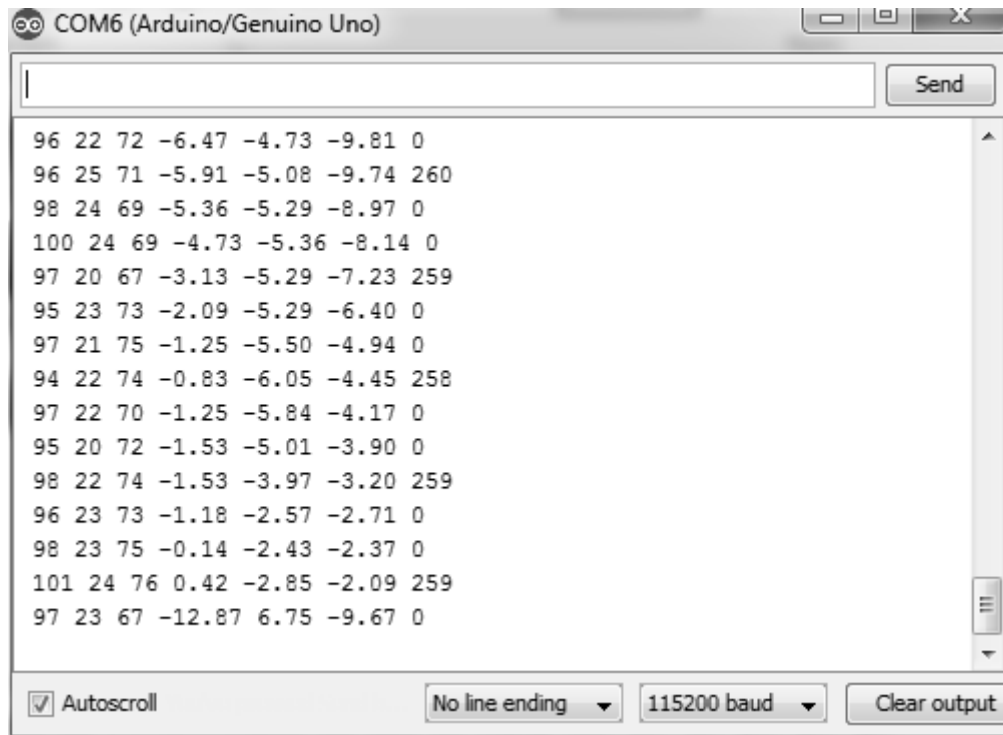
5.7. Thực hiện thu thập dữ liệu

Người dùng cầm thiết bị đo độ dài từ xa (Hình 5.5). Ban đầu đèn laser được bật lên, người dùng chỉ vào điểm cần đo thứ nhất, sau đó nhấn nút record, tiếp theo người dùng chỉ vào điểm cần đo thứ hai, sau đó nhấn nút record để kết thúc việc đọc dữ liệu.



Hình 5.5: Thiết bị đo độ dài từ xa

Dữ liệu được vi điều khiển đọc sau đó truyền qua giao diện giám sát nối tiếp (Serial monitor) tốc độ 115200baud, dữ liệu có dạng (ax, ay, az, gx, gy, gz, distance) với tần số lấy mẫu 100Hz



Hình 5.6: Dữ liệu truyền qua giao diện serial monitor

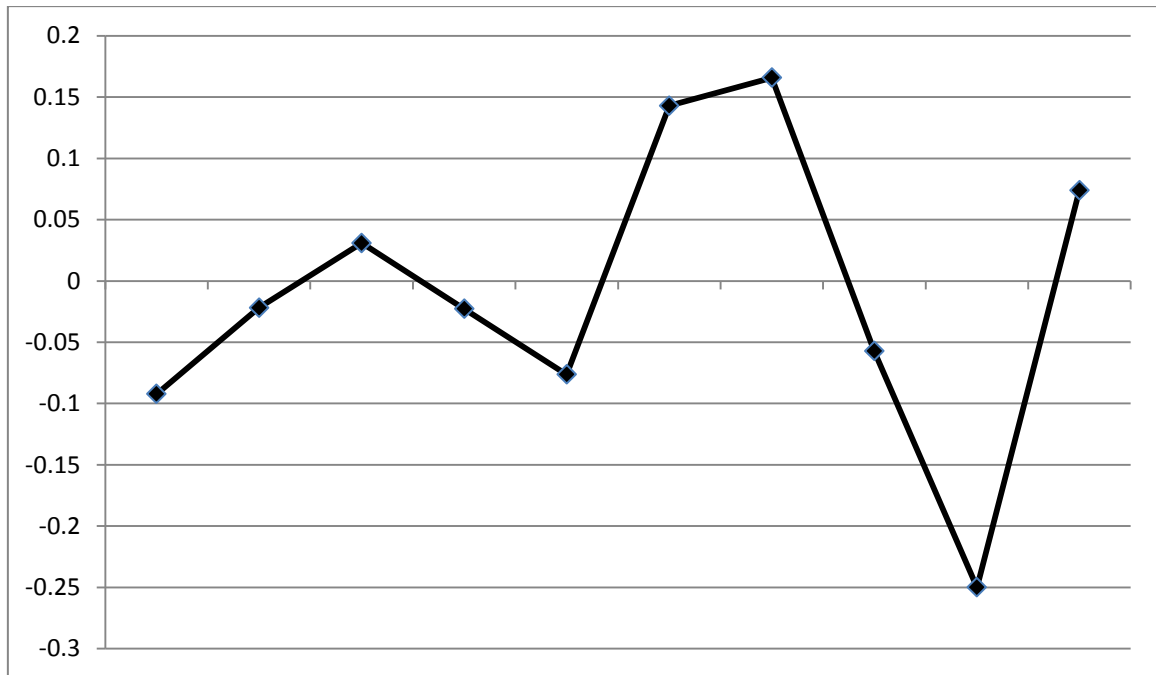
5.8. Thống kê kết quả đo

Tác giả thực nghiệm đo các đoạn AB, BC, AC, mỗi đoạn được thực hiện 10 lần, các vị trí đứng xa là khác nhau, thống kê đo độ dài từ xa được thể hiện dưới bảng sau:

Độ dài (m)	AB		BC		AC	
Giá trị thật	2.03	Sai lệch	2.51	Sai lệch	1.51	Sai lệch
Lần thứ 1	2.1222	-0.0922	2.4385	0.0715	1.4329	0.0771
Lần thứ 2	2.1441	-0.0219	2.4172	0.0213	1.3170	0.1159
Lần thứ 3	2.1132	0.0309	2.5683	-0.1511	1.4665	-0.1495
Lần thứ 4	2.1360	-0.0228	2.5893	-0.021	1.5413	-0.0748
Lần thứ 5	2.2123	-0.0763	2.7328	-0.1435	1.5885	-0.0472
Lần thứ 6	2.0694	0.1429	2.4308	0.302	1.4069	0.1816
Lần thứ 7	1.9037	0.1657	2.4140	0.0168	1.5218	-0.1149
Lần thứ 8	1.9609	-0.0572	2.3843	0.0297	1.3795	0.1423
Lần thứ 9	2.2109	-0.25	2.6866	-0.3023	1.5287	-0.1492
Lần thứ 10	2.1369	0.074	2.7908	-0.1042	1.4317	0.097

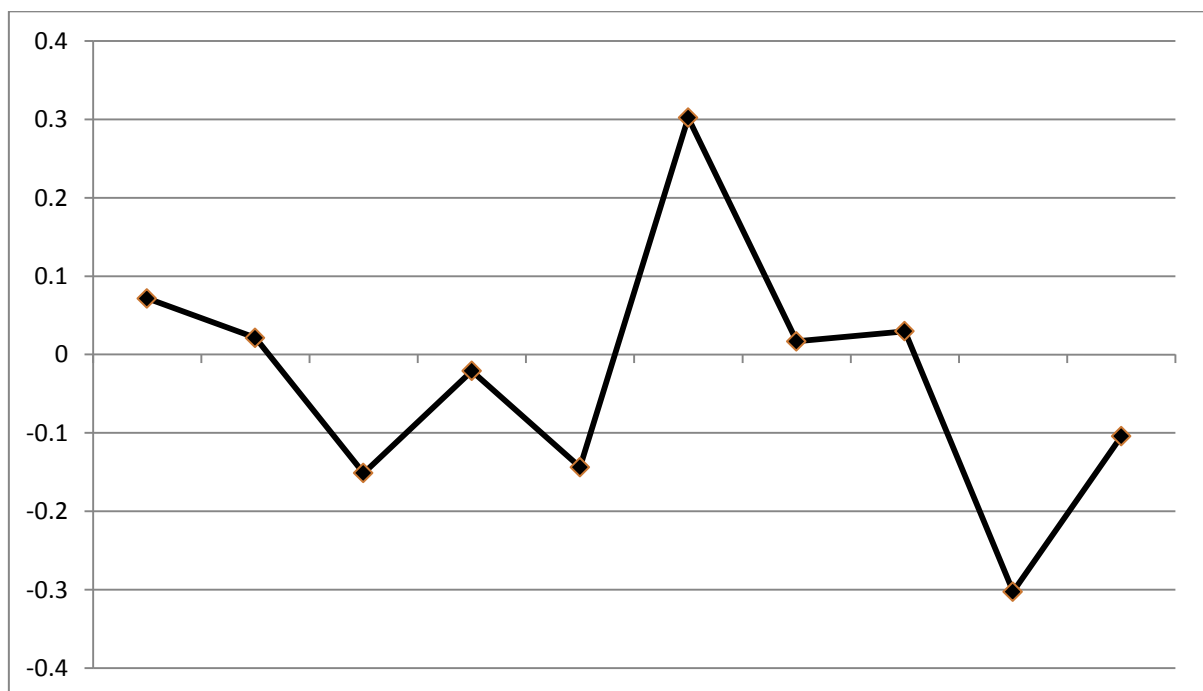
Bảng 5.1: thống kê kết quả đo thực nghiệm

5.9. Đánh giá sai lệch



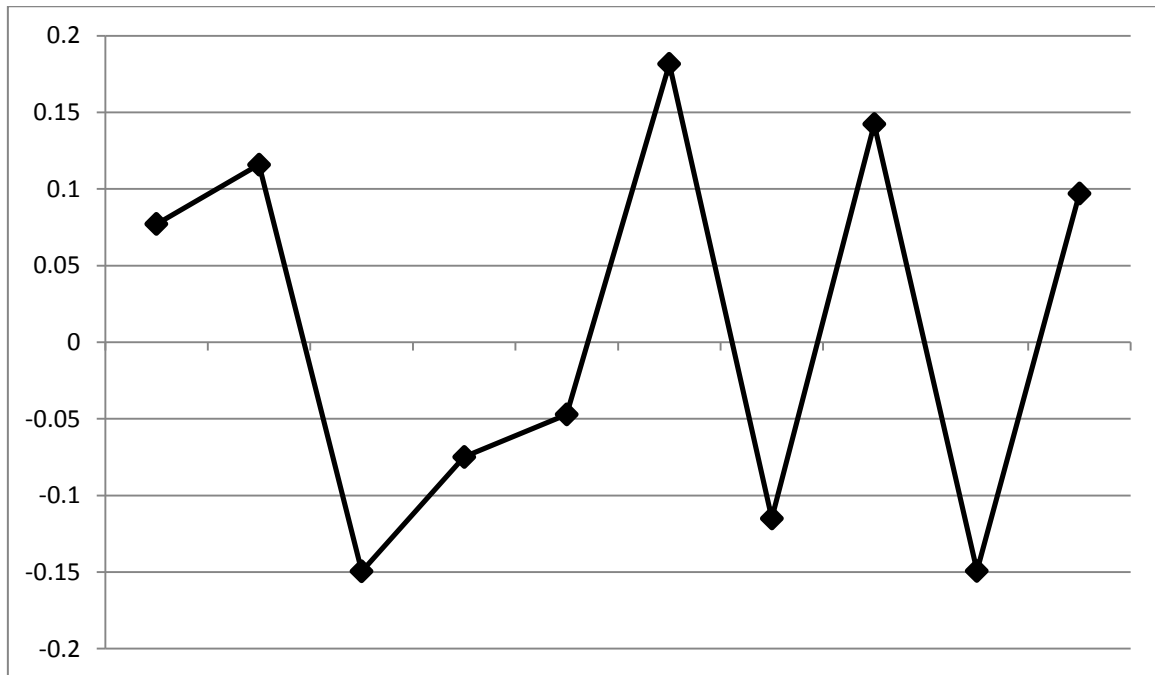
Bảng 5.2: Biểu đồ thống kê sai lệch độ dài AB

Trung bình sai lệch $\bar{y}_{AB} = \frac{1}{n} \sum_{i=1}^n y_{iAB} = -0.01069$



Bảng 5.3: Biểu đồ thống kê sai lệch độ dài đoạn BC

Trung bình sai lệch $\bar{y}_{BC} = \frac{1}{n} \sum_{i=1}^n y_{iBC} = -0.02808$



Bảng 5.4: Biểu đồ thống kê sai lệch độ dài đoạn AC

Trung bình sai lệch $\bar{y}_{AC} = \frac{1}{n} \sum_{i=1}^n y_{iAC} = 0.00783$

Từ bảng 7.1 ta cũng có bảng thống kê sai lệch nhỏ nhất và lớn nhất độ dài đoạn AB, BC, AC như sau:

Đoạn	Sai lệch nhỏ nhất	Sai lệch lớn nhất
L_{AB}	-0.0922 (4.5%)	0.1657 (8.1%)
L_{BC}	-0.3023 (12%)	0.302 (12%)
L_{AC}	-0.1495 (10%)	0.1816 (12.1%)

Bảng 5.5: Bảng thống kê sai lệch nhỏ nhất và lớn nhất từng đoạn AB, BC, AC

Từ bảng 5.2, 5.3, 5.4, 5.5 ta kết luận thiết bị đo xa sai số còn tương đối lớn, chưa đáp ứng được yêu cầu đo độ dài từ xa trong không gian.

5.10. Nguyên nhân sai số

Trong đo lường, các nguyên nhân sai số là sai số hệ thống và sai số ngẫu nhiên, ngoài ra còn có các nguyên nhân sau:

- Cảm biến đo độ dài LIDAR có phạm vi đo lường <40 và sai số $\pm 5\text{cm}$
- Cảm biến quán tính GY-85 là loại cảm biến rẻ tiền hơn gấp nhiều lần cảm biến quán tính chất lượng tốt, ví dụ cảm biến quán tính của hãng Xsens...nên việc đáp ứng về tần số, dữ liệu đọc được có độ chính xác chưa cao.
- Trong quá trình thu thập dữ liệu khi thực hiện việc chỉ lên các điểm cần đo đã có sai lệch
- Quá trình xử lý dữ liệu còn có sai số giữa tần số lấy mẫu của cảm biến đo xa LIDAR và cảm biến quán tính GY-85 khi đồng bộ.

KẾT LUẬN VÀ KIẾN NGHỊ

1. KẾT LUẬN

a. Những mặt đạt được

Về tổng quan luận văn đã hoàn thành trong việc thiết kế thiết bị đo độ dài từ xa ứng dụng cảm biến quán tính.

Ứng dụng được thuật toán định vị quán tính sử dụng bộ lọc Kalman trong ước lượng chuyển động cho cảm biến quán tính để thiết kế thiết bị đo xa, nội dung luận văn đã đề cập đến các vấn đề sau:

- Tìm hiểu về lý thuyết bộ lọc Kalman.
- Tìm hiểu về thuật toán định vị quán tính
- Tìm hiểu cảm biến quán tính
- Cảm biến đo xa LIDAR.
- Phần mềm Matlab xử lý và tính toán dữ liệu từ vi điều khiển
- Thực nghiệm đo độ dài từ xa và so sánh với độ dài trong thực tế

Xây dựng chương trình tính toán chạy trên MATLAB. Từ những vấn đề trên, chúng ta rút ra được một số kết luận sau:

- Với việc sử dụng bộ lọc Kalman và thuật toán định vị quán tính đã đo khoảng cách một cách linh hoạt phục vụ cuộc sống kỹ thuật.
- Thiết bị được thiết kế nhỏ gọn.

b. Hạn chế của đề tài

Mặc dù đã có nhiều cố gắng trong nghiên cứu tìm hiểu nhưng hạn chế về mặt thời gian, độ chính xác của cảm biến quán tính, cũng như kiến thức về học thuật, đề tài này vẫn còn những thiếu sót cần được khắc phục bổ sung như sau:

Thiết bị đo độ dài từ xa đã giải quyết được vấn đề đặt ra tuy nhiên sai số từ kết quả đo từ thiết bị so với độ dài thực tế còn cao.

Việc thu thập dữ liệu có thể mang lại dữ liệu không tốt dẫn đến việc ước lượng độ dài không thực hiện được

Thiết bị đo độ dài từ xa chưa đo được độ dài trực tiếp mà phải thông qua phần mềm Matlab vì vậy việc đo xa chưa thuận tiện và mất nhiều thời gian.

Thiết bị đo được làm để thí nghiệm nên sản phẩm có tính thẩm mỹ chưa cao

2. KIẾN NGHỊ

Với những hạn chế nêu trên tác giả mong muốn có điều kiện sẽ nghiên cứu phát triển, cải thiện đề tài này theo các hướng sau:

Thiết kế thiết bị đo độ dài từ xa đo được độ dài từ xa một cách trực tiếp

Sử dụng thiết bị cảm biến có độ chính xác cao hơn nhằm giảm sai số nhỏ nhất.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Lý thuyết điều khiển phi tuyến, Nguyễn Doãn Phước. NXB KH & KT. 2005.
- [2] Lý thuyết điều khiển nâng cao, Nguyễn Doãn Phước. NXB KH & KT. 2007.
- [3] Matlab & Simulink dành cho kỹ sư điều khiển tự động, *Nguyễn Phùng Quang*, Nhà xuất bản khoa học và kỹ thuật, Hà Nội-2006.
- [4] Hướng dẫn sử dụng cơ bản vi điều khiển Arduino, Học viện hàng không Việt Nam

Tiếng Anh

- [5] Digital signal processing principles algorithms and applications, John G.Proakis, Dimitris G.Manolakis
- [6] Young Soo Suh, “Basic Inertial Navigation Alogorithm for Personal Navigation”, Ulsan University, Korea, 2010.
- [7] Kalman filter, http://en.wikipedia.org/wiki/Kalman_filter
- [8] Vinh Tran-Quang, Phat Nguyen Huu, Takumi Miyoshi, “A Collaborative TargetTracking Algorithm Considering Energy Constraint in WSNs” 15-17 Sept. 2011.
- [9] Y.S. Suh, N.H.Q. Phuong, H.J. Kang, Distance estimation using inertial sensor and vision, Int. J. Control Autom. Syst. 11 (1) (2013) 211–215.
- [10] R. Nevatia, Depth measurement by motion stereo, Comput. Graph. Image Process. 5 (2) (1976) 203–214.
- [11] T. Sugimoto, A. Tohshima, Estimation method for the consciousness level while driving vehicles, Comput. Stand. Interfaces 33 (2) (2011) 136–141
- [12] LIDAR (Light Detection and Ranging Module) Datasheet, URL <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/lidarlite2DS.pdf>.
- [13] Kuipers, J.B. Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality; Princeton University Press: Princeton, NJ, USA, 1999.
- [14] Markley, F.L. Multiplicative vs. Additive Filtering for Spacecraft Attitude Determination. In Proceedings of the 6th Cranfield Conference on Dynamics and Control of Systems and Structures in Space, Riomaggiore, Italy, 18–22 July 2004; pp. 467–474.

PHỤ LỤC 1

Code vi điều khiển viết cho board vi điều khiển Arduino UNO R3 như sau:

```
#include "GY_85.h"
#include <Wire.h>
#include "I2C.H"
#include "TimerOne.H"
#include "UART_ARDUINO.H"

#define LIDARLite_ADDRESS 0x62 // Default I2C Address of LIDAR-Lite.
#define RegisterMeasure 0x00 // Register to write to initiate ranging.
#define MeasureValue 0xFF // Value to initiate ranging.
#define RegisterHighLowB 0x8f // Register to get both High and Low bytes in 1
call.

#define offsetX -123 // OFFSET values
#define offsetY -16
#define offsetZ -10

#define gainX 128
#define gainY 128
#define gainZ 128

int ax;
int ay;
int az;
float aclx;
float acly;
float aclz;
float gx;
float gy;
```

```

float gz;
float gt;
int distance=0;
int dem=0;
volatile int state = LOW;
int check=0;
const int ledPin = 5;

byte distanceArray[2]; // array to store distance bytes from read
Lệnh sử dụng trong Matlab:
GY_85 GY85; //create the object
void setup()
{
    I2c.begin(); // Opens & joins the i2c bus as master
    delay(100); // Waits to make sure everything is powered up before sending or receiving data
    I2c.setTimeout(5); // Sets a timeout to ensure no locking up of sketch if I2C communication fails

    Wire.begin();
    delay(10);
    Serial.begin(115200);
    delay(10);
    GY85.init();
    delay(10);
    // Serial.print ( "accelerometer" );
    pinMode(5, OUTPUT);
    // initialize the pushbutton pin as an input:
    pinMode(3, INPUT);
    Timer1.initialize(10000); // initialize timer1, and set a 1/2 second period
    Timer1.attachInterrupt(turnon); // attaches callback() as a timer overflow interrupt
}
void loop()

```

```

{
  if (digitalRead(3) == 0) {
    delay(20);
    if (digitalRead(3) == 0)
    {
      state = !state;
      digitalWrite(ledPin, state);
      while(digitalRead(3) == 0);
    }
  }
  if(check)
  {
    ax = GY85.accelerometer_x( GY85.readFromAccelerometer() );
    ay = GY85.accelerometer_y( GY85.readFromAccelerometer() );
    az = GY85.accelerometer_z( GY85.readFromAccelerometer() );
    //aclx=(float)ax/gainX*9.8;
    //acly=(float)ay/gainY*9.8;
    //aclz=(float)az/gainZ*9.8;

    //int cx = GY85.compass_x( GY85.readFromCompass() );
    //int cy = GY85.compass_y( GY85.readFromCompass() );
    //int cz = GY85.compass_z( GY85.readFromCompass() );

    gx = GY85.gyro_x( GY85.readGyro() );
    gy = GY85.gyro_y( GY85.readGyro() );
    gz = GY85.gyro_z( GY85.readGyro() );

    Serial.print ( " " );
    Serial.print ( ax );
    Serial.print ( " " );
    Serial.print ( ay );

```



```

Serial.print ( " ");
  Serial.print ( az );
  Serial.print ( " ");
  Serial.print ( gx );
  Serial.print ( " ");
  Serial.print ( gy );
  Serial.print ( " ");
  Serial.print ( gz );
  Serial.print ( " ");
  check=0;
if(dem==2)
{
  // Read 2byte distance from register 0x8f
  uint8_t nackack = 100; // Setup variable to hold ACK/NACK responses
  //nackack = 100; // Setup variable to hold ACK/NACK responses
  //while (nackack != 0){ // While NACK keep going (i.e. continue polling until success
  message (ACK) is received )
    nackack = I2c.read(LIDARLite_ADDRESS,RegisterHighLowB, 2, distanceArray); //
  Read 2 Bytes from LIDAR-Lite Address and store in array
    //delay(1); // Wait 1 ms to prevent overpolling
  // }
  distance = (distanceArray[0] << 8) + distanceArray[1]; // Shift high byte [0] 8 to the left
  and add low byte [1] to create 16-bit int
    Serial.println(distance);
    distance=0;
  }
else {Serial.println(distance);
}
}
}
void turnon()
{

```

```

if (state==HIGH) {
    check=1;
    dem++;
    if(dem==3)
    { dem=0;
      uint8_t nackack = 100; // Setup variable to hold ACK/NACK responses
      // while (nackack != 0){ // While NACK keep going (i.e. continue polling until success
      message (ACK) is received )
      nackack = I2c.write(LIDARLite_ADDRESS,RegisterMeasure, MeasureValue); // Write
      0x04 to 0x00
      //delay(1); // Wait 1 ms to prevent overpolling
      //}
    }
    //Serial.println(distance);
    Timer1.initialize(10000);      // initialize timer1, and set a 1/2 second period
    Timer1.attachInterrupt(turnon); // attaches callback() as a timer overflow interrupt
  }
}

```

PHỤ LỤC 2

Để chạy mô phỏng và tính toán khoảng cách tác giả đề xuất các bước làm sau:

Bước 1:

Setpath thư mục thư viện chứa các hàm, công thức của Eule, Quaternion...Thư mục này tác giả đặt tên “mymatlab”, lưu ý chọn thêm option Sub folder để tất cả các tập tin trong thư mục đều được thêm vào.

Bước 2:

Tìm ma trận nhiễu. Để tìm ma trận nhiễu ta thực hiện như sau:

Ta sử dụng mô hình thí nghiệm đã làm chúng ta lấy như cách lấy dữ liệu bình thường, nhưng ta không di chuyển mô hình. Lúc này dữ liệu xuất ra giao diện serial monitor là ma trận nhiễu. Tác giả đề xuất đặt tên tập tin là “bias.mat”. Ma trận “bias.mat” cũng được Setpath.

Bước 3:

Tạo tập tin dữ liệu có phần mở rộng .m, đặt tên tập tin là get_data.m

Code viết cho tập tin này như sau:

```
clear all
clc
load 'đường dẫn\data.mat'
a(:,1)=a(:,1)*9.8/4
a(:,2)=a(:,2)*9.8/4
a(:,3)=a(:,3)*9.8/4

a(:,4)=a(:,4)/180*pi
a(:,5)=a(:,5)/180*pi
a(:,6)=a(:,6)/180*pi
save(đường dẫn\data_edit.mat,'a');
```

Bước 4:

Tạo tập tin tính toán khoảng cách có phần mở rộng .m, đặt tên tập tin là distance.m.

Code viết cho tập tin này như sau:

```
clear all
clc
close all
ghat = 9.810;
%load bais
load('bias.mat');
```

```

temp=a;
ya0=temp(:,1:3)';
yg0=temp(:,4:6)';
temp=mean(yg0');
bghat=temp';
bahat=zeros(3,1);
T=0.01;
Qba0 = 0.000001 * eye(3);
Qbg0 = 0.000001 * eye(3);
%eye=ma tran don vi
load(['data_edit.mat']);
temp=a(100:length(a),:);
% ya=[];
ya=temp(:,1:3)';
yg=temp(:,4:6)';
% ym=temp(:,7:9)';
N=max(size(ya));
% zero velocity interval
ygnorm = zeros(1,N);
yanorm = zeros(1,N);
yadiff = zeros(1,N);

for i = 1:N
    ygnorm(i) = norm(yg(:,i));
    yanorm(i) = norm(ya(:,i));
    if ( i > 1 )
        yadiff(i) = norm( ya(:,i) - ya(:,i-1) );
    end
end
[zerog] = find_interval_lessthan_edit(ygnorm,0.2,40);
[zeroa] = find_interval_lessthan_edit(yadiff,0.45,40);

zerovel = 1 * zerog .* zeroa;
% zerovel=zeros(1,N);
up=[];
n=1;
down=[];
m=0;
for i=2:N-1
    if zerovel(i)>zerovel(i-1)
        n=n+1;
        up(n)=i;
    end
end

```

```

    if zeroval(i)>zeroval(i+1)
        m=m+1;
        down(m)=i;
    end
end
up(1)=2;
down(length(down)+1)=N;
% distance correction
for k=1:1
    load('LIDAR_distance_calib.mat')
m=0;
count=0
dis_vector=zeros(1,N);
dis_value=zeros(1,N);
dis_value=a(:,7);
for i=1:length(dis_value)
    if dis_value(i)>0
        dis_vector(i)=1;
    end
    dis_value(i)=dis_value(i)/100;
end
for k=1:length(up)
    dis_mean(k)=sum(dis_value(up(k):down(k)))/sum(dis_vector(up(k):down(k)));
end
floorzero=0;
ym=ya;
[r,v,q4,eulercom4,bghat,bahat,Qbg0,Qba0,zeroval] =
personal_navigation0(ya,yg,ym,floorzero,zeroval,ghat,bghat,bahat,Qbg0,Qba0);
len = norm(r(:,length(ya)))
figure
plot3(r(1,:),r(2,:),r(3,:))
axis equal
x=[0.9994 ; 0.0135 ; 0.0311];
% Lidar point
b1=quaternion2dcm(mean(q4(:,1),2))'*(dis_mean(1))*x
b2=quaternion2dcm(mean(q4(:,up(2):down(2)),2))'*(dis_mean(2))*x+mean(r(:,up(2):down(
2)),2)
b=[b1 b2 ]
quaternion2euler(q4(:,1))*180/pi
quaternion2euler(mean(q4(:,up(2):down(2)),2))*180/pi
%Length
len=[norm(b1-b2)]
end

```


QUYẾT ĐỊNH

Về việc giao đề tài và trách nhiệm của người hướng dẫn luận văn thạc sĩ

HIỆU TRƯỞNG TRƯỜNG ĐẠI HỌC BÁCH KHOA

Căn cứ Nghị định số 32/CP ngày 04 tháng 4 năm 1994 của Chính phủ về việc thành lập Đại học Đà Nẵng;

Căn cứ Thông tư số 08/2014/TT-BGDĐT ngày 20 tháng 3 năm 2014 của Bộ trưởng Bộ Giáo dục và Đào tạo về việc ban hành Quy chế tổ chức và hoạt động của đại học vùng và các cơ sở giáo dục đại học thành viên; Quyết định số 6950/QĐ-DHĐN ngày 01 tháng 12 năm 2014 của Giám đốc Đại học Đà Nẵng về việc ban hành Quy định nhiệm vụ, quyền hạn của Đại học Đà Nẵng, các cơ sở giáo dục đại học thành viên và các đơn vị trực thuộc;

Căn cứ Thông tư số 15/2014/TT-BGD&ĐT ngày 15 tháng 5 năm 2014 của Bộ trưởng Bộ Giáo dục và Đào tạo về việc ban hành Quy chế Đào tạo trình độ Thạc sĩ; Quyết định số 598/QĐ-DH BK ngày 27/12/2016 của Hiệu trưởng trường Đại học Bách khoa về việc ban hành Quy định đào tạo trình độ thạc sĩ;

Căn cứ Quyết định số 1163/DH BK-DT ngày 25/08/2016 của Hiệu trưởng trường Đại học Bách Khoa về việc công nhận học viên cao học trúng tuyển;

Căn cứ Tờ trình số 62/KD của khoa Điện về việc ra Quyết định giao đề tài và người hướng dẫn luận văn thạc sĩ cho học viên cao học chuyên ngành Kỹ thuật điều khiển và tự động hóa;

Xét đề nghị của Trưởng Phòng Đào tạo,


QUYẾT ĐỊNH

Điều 1. Giao cho học viên cao học **Nguyễn Như Thế**, lớp K33.TĐH.DL, chuyên ngành Kỹ thuật điều khiển và tự động hóa, thực hiện đề tài luận văn "*Thiết kế thiết bị đo độ dài từ xa ứng dụng cảm biến quán tính*", dưới sự hướng dẫn của TS. Nguyễn Anh Duy, Trưởng Cao đẳng Công nghệ - Đại học Đà Nẵng.

Điều 2. Học viên cao học và người hướng dẫn có tên ở Điều 1 được hưởng các quyền lợi và thực hiện nhiệm vụ theo đúng quy chế đào tạo thạc sĩ hiện hành của Bộ Giáo dục và Đào tạo, quy định đào tạo thạc sĩ của Trường Đại học Bách khoa.

Điều 3. Các ông/bà Trưởng Phòng Đào tạo, Trưởng phòng Kế hoạch - Tài chính, Trưởng khoa Điện, người hướng dẫn luận văn và học viên có tên ở Điều 1 căn cứ Quyết định thi hành.

Nơi nhận:
- Như điều 3;
- Lưu: Phòng DT.

KT. HIỆU TRƯỞNG
PHÓ HIỆU TRƯỞNG

PGS.TS. Lê Thị Kim Oanh